MGCP 媒体网关控制协议

MGCP MEDIA GATEWAY CONTROL PROTOCOL

（送审稿）

2003 年 12 月 31 日发布　　2003 年 12 月 31 日实施

中国电信集团公司发布

前言

媒体网关控制协议 MGCP 是下一代网络中的接口协议之一，它应用于下一代网络中媒体处理和信令控制分离后所产生的控制接口。本标准以 IETF 的相关标准为基础，结合中国电信网络的实际情况，综合中国电信集团公司对下一代网络的研究实验成果而制定。它是中国电信在下一代网络建设中引进、测试和研发软交换、媒体网关控制器、媒体网关、媒体服务器等相关设备的规范和依据。

本标准的附录都是标准附录。

本标准由中国电信集团公司提出。

本标准由中国电信集团公司归口。

本标准 2003 年 12 月 31 日首次发布。

本标准由中国电信集团公司负责解释。

1 范围

本标准规定了媒体网关控制设备（媒体网关控制器/软交换设备）和相应的媒体处理设备（网关/媒体服务器/IP 智能终端等）之间采用 MGCP 协议进行通信时的协议要求。

本标准适用于媒体网关、媒体网关控制器和软交换设备的研制、开发和引进。

2 规范性引用文件

RFC3435（2003.01）	网关控制协议
RFC 2805	媒体网关控制协议结构和要求
RFC 1819	RTP 协议
RFC 2327	SDP 协议
RFC 2401	IP 协议安全机制
RFC 2402	AH 协议
RFC 2406	ESP 协议

3 定义及缩略语

本协议用到以下定义：

媒体网关（MG）：MG 将一种网络中的媒体转换成另一种网络所要求的媒体格式。例如：MG 能够完成电路交换网的承载通道和分组网的媒体流之间的转换。MG 可以处理音频、视频或者 T.120，也可以具备处理这三者任意组合的能力。MG 能够进行全双工的媒体转换，MG 可以播放视频/音频消息，实现其它 IVR 功能，也可以进行媒体会议。

呼叫代理/媒体网关控制器（CA/MGC）：CA/MGC 对 MG 中与媒体通道连接控制相关的呼叫状态进行控制。

中继（Trunk）：两个交换系统间的一个通信通道，例如：T1 或 E1 中的一个时系 DS0。

端点（Endpoint）：发送或接收数据的端点

连接（Connection）：呼叫代理基于端点控制 MG 建立，提供端点收/发数据分组媒体资源

命令（Command）：本协议定义了一些命令用于对协议连接模型中的逻辑实体(端点和连接)进行操作和管理。

事务（Transaction）：由一个命令和对它的响应构成事务。

通配值（Wildcard）：协议语法中采用的特殊符号，有“ CHOOSE ”和“ ALL ”两种。“ ALL ”表示需要使用所有满足条件的取值，“ CHOOSE ”表示需要选择一个满足条件的取值。在没有特殊说明时，通配值往往特指“ ALL ”。

本协议采用以下缩略语：

MGCP：媒体网关控制协议

SDP：会话描述协议

RTP：实时传输协议

4 连接模型

MGCP 制定了这样一个连接模型，它的基本构件是端点（Endpoint）和连接（Connection）。连接又组合为呼叫，一个或多个连接可以属于一个呼叫，连接和呼叫可以通过一个或多个呼叫代理发起建立。

4.1 端点

发送或接收数据的端点。

4.1.1端点的类型

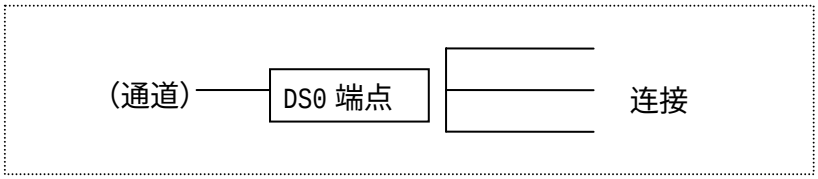
假定媒体网关支持各类端点的集合，端点的类型决定了网关的功能性。可以抽象出以下基本端点类型：

- 数字通道（DS0）
- 模拟线路
- 录音通知服务器接入点
- 交互式语音应答（IVR）接入点
- 会议桥接入点
- 包中继
- ATM 中继侧接口

本节将介绍这些端点可预期的行为。需要说明的是，本列表不是最终列表，将来还可能定义其他类型的端点，例如，用于检测网络质量的测试端点，或者在帧中继的虚电路上管理语音信道复用的帧中继端点。

4.1.1.1 数字通道（DS0）

数字通道提供 64Kbps 业务，中继以及 ISDN 接口中就包含数字通道。它们通常作为构成数字多路复用传输（比如 T1，E1，T3 或 E3 接口）系统的一部分。支持数字信道的媒体网关能够翻译信道上传来的数字信息，这些数字信息可以采用 A 律或μ律，使用每个样本 8 比特或 7 个比特的方式进行语音包编码。当媒体网关还支持 NAS 网络接入服务时，网关必须有能力接收音频编码数据（modem 连接）或二进制数据（ISDN 连接）并将其转换为数据包。

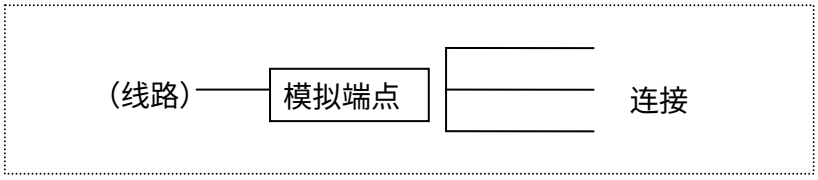


媒体网关能够在端点和分组网之间，或者在同一个网关内部的一个端点和其它端点之间建立多个连接。源自这些连接的信息必须根据连接模式（mode）进行混合处理，本标准在后面将详细对此加以介绍。一个端点支持的连接的确切数目是网关的一个特性，在实践中，会随着网关的资源配置而变化。

在某些情况下，数字通道也用来传输信令，比如，七号信令（SS7）中的 F-链路，或 ISDN 的 D-信道；支持这种信令功能的媒体网关应能够从呼叫代理接收和发送信令包，具体方法是，使用 IETF 的 SIGTRAN 工作组定义的“ back haul ”过程。有时，数字通道也与随路信令配合使用，比如“ MF-R2 ”；支持这种信令功能的媒体网关应能够检测和生成相应的信令，比如“ wink ”或“ A ”，具体过程可参见本标准中定义的事件信令及其报告程序。

4.1.1.2 模拟线路

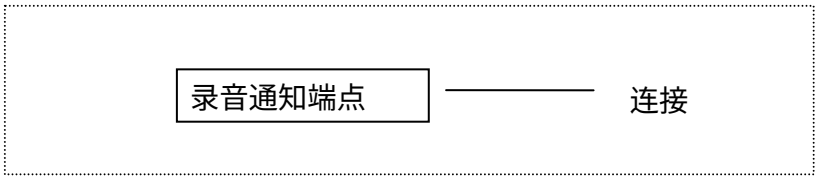
模拟线路可以作为“ 用户 ”接口，为传统的电话提供服务；它也可以作为“ 业务 ”接口，允许网关收发模拟呼叫。当媒体网关还支持 NAS 网络接入服务时，网关应能接收音频编码数据（modem 连接）并将其转换为数据包。



媒体网关能够在端点和分组网之间，或者在同一个网关内部的一个端点和其它端点之间建立多个连接。源自这些连接的信号必须根据连接模式（mode）进行混合处理，本标准在后面将详细对此加以介绍。一个端点支持的连接的确切数目是网关的一个特性，在实践中，会随着网关的资源配置而变化。然而，对一个典型的网关来说，它的每个端点应该能够支持两个或三个连接，以便能够提供像“ 呼叫等待 ”或“ 三方通话 ”之类的业务。

4.1.1.3 录音通知服务器接入点

录音通知服务器端点提供录音通知业务的接入。在呼叫代理的请求下，录音通知服务器将播放指定的录音通知。来自呼叫代理的录音通知请求要符合本标准关于事件信令及其报告过程的规定。

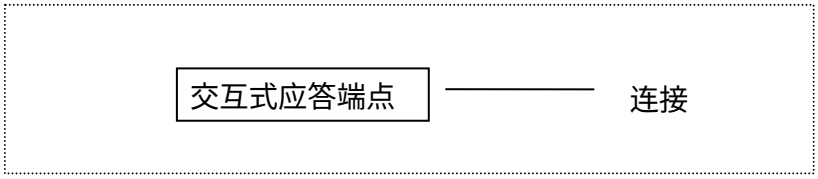


不要求一个特定的录音通知端点同时支持多个连接。如果多个连接建立在相同的端点上，将在所有连接上同时播放相同的录音通知。

一般来说，到录音通知服务器的连接都是单向的，或“ 半双工 ”的 —— 不期望录音通知服务器接听来自连接的音频信号。

4.1.1.4 交互式应答接入点

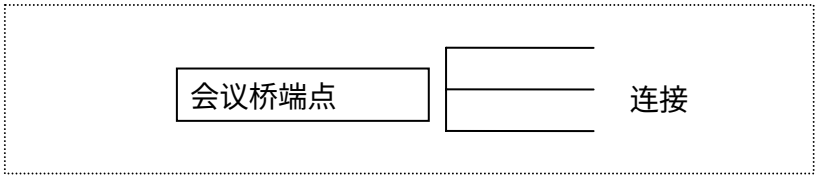
交互式语音应答(IVR)端点提供了对 IVR 业务的接入。在呼叫代理的请求下，IVR 服务器将“ 播放 ”录音通知或通知音，并能接听来自用户的响应，例如，DTMF 输入或声音消息等。来自呼叫代理的请求要符合本标准中关于事件信令及其报告过程的规定。



不要求一个特定的 IVR 端点同时支持多个连接。如果多个连接建立在相同的端点上，将在所有连接上同时播放相同的通知音和录音通知。

4.1.1.5 会议桥接入点

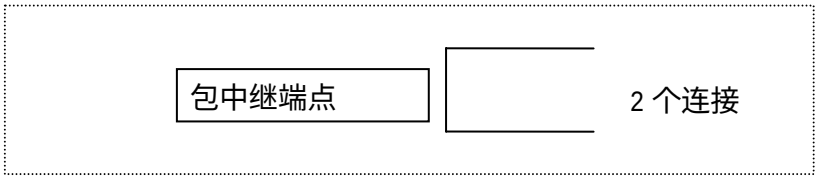
会议桥端点用于为特定会议提供接入。



媒体网关能够在端点和分组网之间，或者在同一个网关内部的一个端点和其它端点之间建立多个连接。源自这些连接的信息必须根据连接模式（mode）进行混合处理，本标准在后面将详细对此加以介绍。一个端点支持的连接的确切数目是网关的一个特性，在实践中，会随着网关的资源配置而变化。

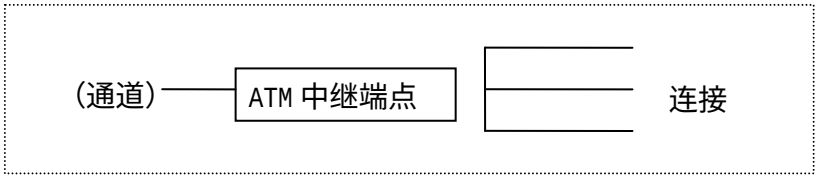
4.1.1.6 包中继

包中继端点是会议桥的特殊形式，典型地，它只支持两个连接。包中继可能位于公共网络与受保护网络之间的防火墙上，或为不兼容网关提供互通功能的代码转换服务器上，例如，不支持兼容压缩算法的网关，或在不同传输网络（如 IP 和 ATM）间运行的网关。



4.1.1.7 ATM“ 中继侧 ”接口

典型地，ATM“ 中继侧 ”端点可见于当一个或多个 ATM 永久虚电路用于替代传统的“ TDM ”中继电路交换时。当用 ATM/AAL2 时，多个中继或通道复合在一个虚电路上，这些中继中的每一个都对应于一个端点。



媒体网关能够在端点和分组网之间，或者在同一个网关内部的一个端点和其它端点之间建立多个连接。源自这些连接的信息必须根据连接模式（mode）进行混合处理，本标准在后面将详细对此加以介绍。一个端点支持的连接的确切数目是网关的一个特性，在实践中，会随着网关的资源配置而变化。

4.1.2端点标识符

端点标识由不区分大小写字符的两部分组成，它们是：

- 管理此端点的网关域名（domain-name）
- 网关内部的一个本地端点名（local-endpoint-name）

端点名称的形式如下： local-endpoint-name@domain-name

其中，域名（domain-name）是一个符合 RFC1034 定义的绝对域名，且包含主机部分。例如：

mygateway.whatever.net；

此外，域名也可以是符合 RFC821 规定的 IP 地址形式。例如：

[192.168.1.2]

IPv4 和 IPv6 地址都可以使用，但是通常不主张在端点标识符中使用 IP 地址。

注意，由于域名是端点标识符的组成部分，关于相同实体的域名的不同形式或不同值是不能随意互换的。必须总是使用最近提供的格式和值。

本地端点名的语法是分层结构的，其中，最笼统的部分在最左边的条目中，而最明确的部分在最右边的条目中。精确的语法与需要命名的端点类型有关，可以用指示端点类型的标识开头。无论怎样，本地端点名都必须遵循以下命名规则：

1) 命名路径的各个条目之间必须用符号斜杠分开（“ / ”，十六进制编码“ 2F ”)。

2) 每个条目均是由字母、数字或其它可打印字符组成的字符串，但分隔符("/", "@")、通配符("*", "$")以及空格符除外。

3) 命名路径中的条目有时需要使用通配符，通配符可以用符号“ * ”和 “ $ ”表示。因此，如果完整的本地端点名为以下形式：

term1/term2/term3

那么，每个条目（term）都可以是通配符：

如果 term1 是通配符，表示为*/term2/term3

如果 term2 是通配符，表示为 term1/*/term3

如果 term3 是通配符，表示为 term1/term2/*

如果 term2 和 term3 是通配符，表示为 term1/*/*

以上每一种情况下，都可以“ $ ”符号替换“ * ” 符号，代表不同的含义。

4) 用“ * ”表示的条目（term）意思是“ 使用此条目在媒体网关范围内已知的所有取值 ”。除非另外指明，它涉及到所有的做过业务配置的端点，并和他们实际的业务状态无关，例如，在服务中或退出服务。

5) 用“ $ ”表示的条目（term）的意思是“ 使用此条目在媒体网关范围内已知的任一取值 ”。除非另外指明，它仅涉及到在服务状态（in-service）中的端点。

此外，建议呼叫代理遵循以下规定：

- 通配符应该从右边开始使用，因此，如果一个条目是通配符，那么该条目右边的所有条目也应该是通配符。
- 在混合使用“ * ”与“ $ ”通配符时，“ $ ”符号应该从右边使用，因此，如果一个条目包含“ $ ”符号，那么该条目右边的所有条目也应该包含“ $ ”符号。

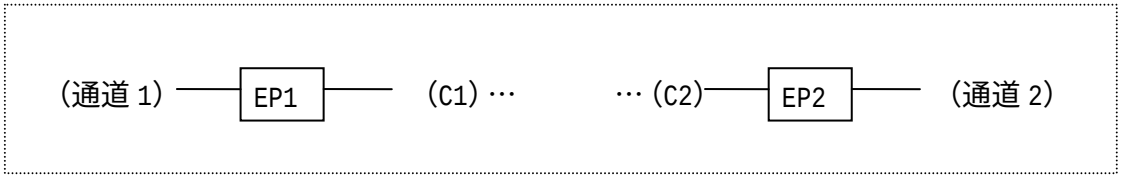
特定命令的描述可以增加进一步的使用通配符的标准，上述作为一般通用规则。

需要注意的是，当两种通配符分别应用在不同条目中时，它们应该按照从左到右的顺序来处理。例如，如果端点名称为“ a/1 ”、“ a/2 ”、“ b/1 ”和 “ b/2 ”。那么，“ $/* ”（不推荐这样使用）将会解释为“ a/1,a/2 ”或“ b/1,b/2 ”;然而，“ */$ ”将会解释为“ a/1,b/1 ”、“ a/1,b/2 ”、“ a/2,b/1 ”或“ a/2,b/2 ”。在命令中混合使用通配符容易产生错误，因而不主张混合用法。

仅由一个通配符组成的本地端点名被认为是媒体网关内的所有（*）或任意（$）端点。

4.2 呼叫和连接

连接由呼叫代理基于涉及呼叫的每一个端点上建立。典型的例子是，两个 DS0 端点（EP1 和 EP2）之间存在连接，那么，控制这两个端点的呼叫代理会建立起两个连接（C1 和 C2）：



每个连接分配一个由端点指定、本地唯一的连接标识符，连接的特性由连接属性来表征。

当两个端点所在的网关由同一个呼叫代理控制时，可以通过以下三个步骤创建连接：

1) 呼叫代理要求一个网关在第一个端点上“ 建立一个连接 ”（CRCX 命令）。网关为连接分配资源，并在对命令的响应中提供“ 会话描述（SDP） ”信息。在“ 会话描述 ”中包含了第三方向新建立的连接发送数据包所必需的信息，例如，IP 地址、UDP 端口以及编解码参数。

2) 呼叫代理要求另一个网关在第二个端点上“ 建立一个连接 ”（CRCX 命令）。命令中包含第一个网关提供的“ 会话描述 ”信息。网关为第二个连接分配资源，并在响应消息中提供自己的“ 会话描述 ”信息。

3) 呼叫代理向第一个端点发送“ 修改连接（MDCX） ”命令，命令中包含第二个端点的“ 会话描述 ”信息。修改连接完成后，在两个端点之间就可以进行双向通信了。

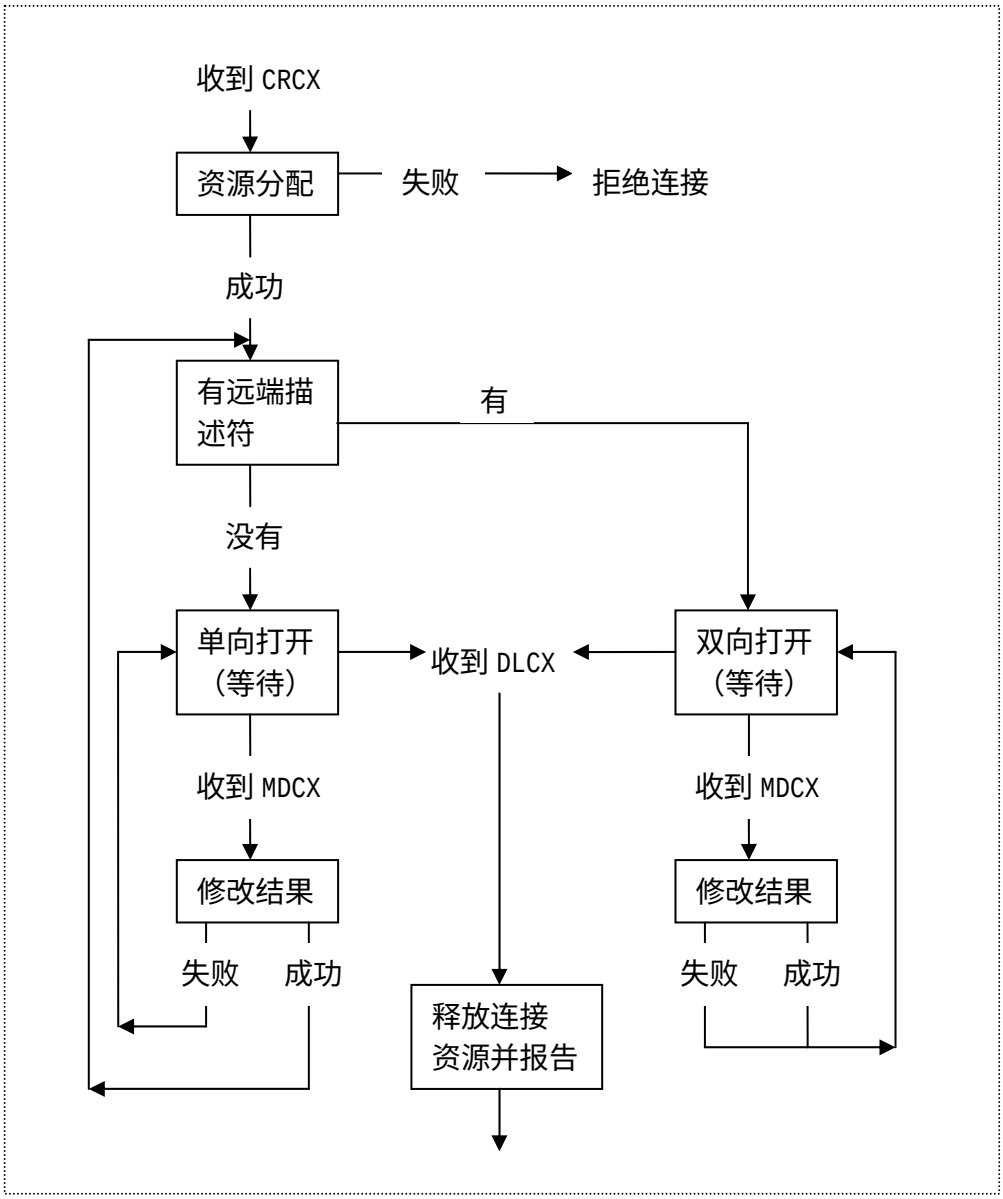
当两个端点分别处于不同的呼叫代理控制的网关内时，两个呼叫代理之间将通过呼叫代理间的通信协议来交换信息，以此来同步两个端点的连接建立。

一旦建立连接，连接参数可以通过“ 修改连接（MDCX） ”命令随时修改。例如，呼叫代理可以命令网关修改某个连接使用的编解码算法；或者在连接被“ 重定向 ”时，修改数据发送所使用的 IP 地址以及 UDP 端口号。

呼叫代理通过向网关发送“ 删除连接（DLCX） ”命令来删除连接。某些具体情况下，网关也可

以通过该命令来通知呼叫代理的某个连接无法保持。

下图从网关的角度，提供一个连接的状态示意图：



4.2.1呼叫标识符

每个连接的属性之一就是呼叫标识符（Call Identifier），就 MGCP 协议而言，呼叫标识符几乎没有语义上的内涵，主要是考虑到后向兼容才加以保留。

呼叫由唯一的标识符指定，与其下的传输平台和代理无关。呼叫标识符是十六进制字符串，由呼叫代理生成，最大长度为 32 字符。

呼叫标识符最好在系统内是唯一的，至少，在控制相同网关的所有呼叫代理中是唯一的，由此，从网关来看，呼叫标识符是唯一的。当呼叫代理在相同或不同的网关内建立属于同一呼叫的多个连接时，属于相同呼叫的所有连接应该共享相同的呼叫标识符。呼叫标识符可以用于计费和管理过程，这超出了本标准所讨论的范畴。

4.2.2连接标识符

连接标识符（connectionID）在网关收到一个建立连接请求时由网关生成，它标识了一个端点相关的连接。在 MGCP 协议中，连接标识符按照十六进制字符串来处理。网关必须确认一个适当的等待周期，至少 3 分钟，在此期间，已经结束的连接所使用的连接标识符不被用于相同端点上的新建连接上（网关可以决定在其范围内使用具有唯一性的标识符）。连接标识符的最大长度是 32 字符。

4.2.3资源管理与连接特性

和连接有关的资源有很多种，比如特定的信号处理功能或打包功能。一般来说，这些资源分为两类：

1) 外部可见资源，它影响到“网络上的比特（the bits on the network）”格式，必须通知连接相关的第二个端点。

2) 内部资源，确定在连接上发送哪个信号以及端点如何处理收到的信号。

对连接的资源分配，或更一般地说，对连接的处理，都由网关按照来自呼叫代理的指令进行选择。呼叫代理通过发送两类参数来向网关提供这些指令：

本地指令：指示网关选择应该用于一个连接的资源。

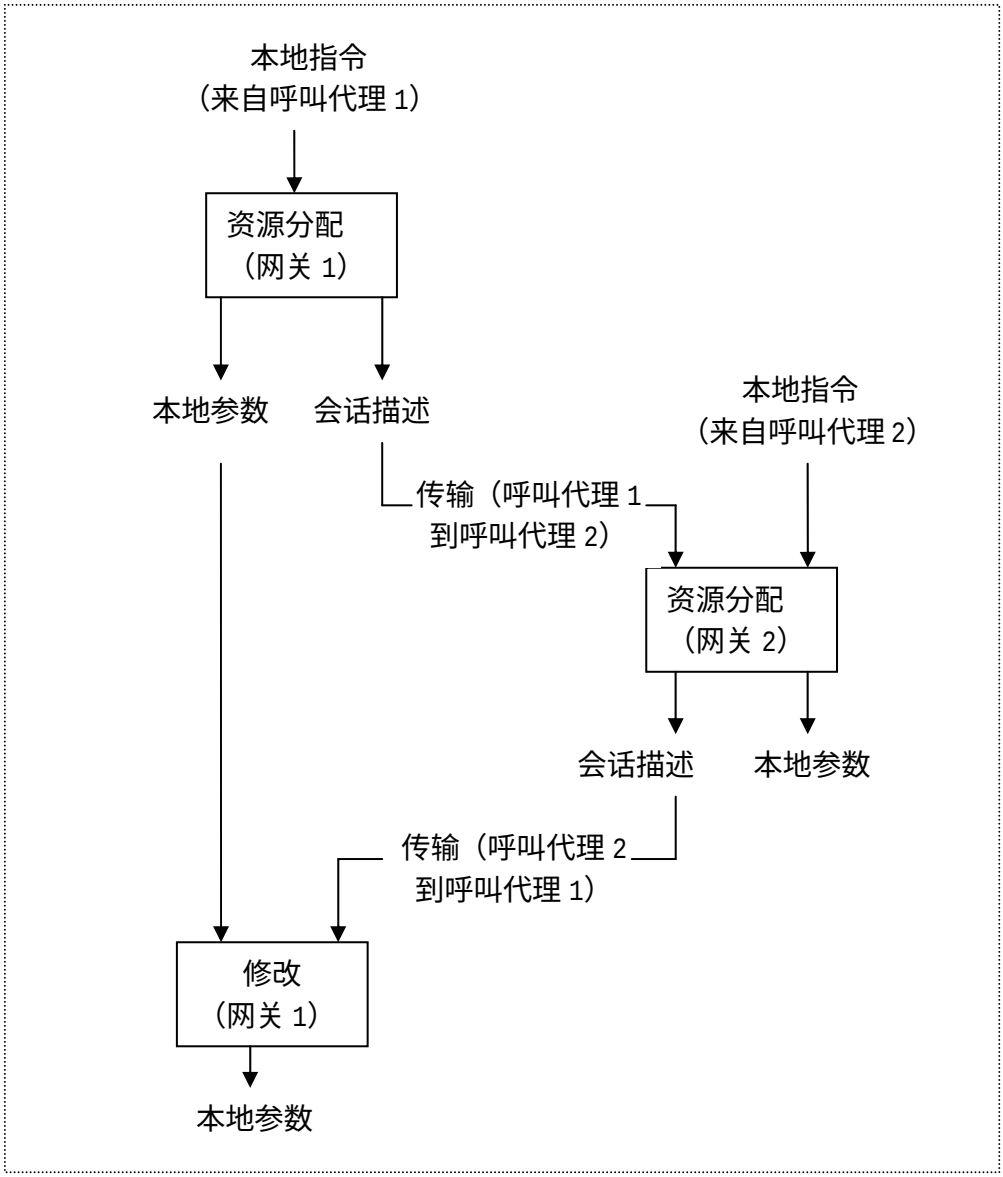
会话描述：当提供时，是指连接的另一端提供的会话描述信息（也叫远端会话描述）。

本地指令定义了一些诸如连接模式（如只发、收发）、推荐的编码或打包算法、回声抑制或静音压缩的使用等参数（详细的列表参见建立连接命令 CRCX 中的本地连接选项参数 LocalConnectionOptions）。根据参数的不同，呼叫代理可以指定一个值、一个取值范围或者不指定值，这样就允许实现不同级别的控制，从由呼叫代理来指定连接处理的微小细节的严格控制方式，到呼叫代理只指定范围（如最大带宽），而让网关依据该范围选择具体取值的宽松控制方式。

基于本地指令的参数值，网关决定分配给连接的资源。在得到了远端会话描述的情况下，网关将选择远端会话描述中的值，但并不是绝对要求本地会话参数要和远端会话参数相同。

一旦分配了资源，网关将生成一个描述收发包方式的会话描述（session description）。注意，会话描述符有时提供一个取值范围。例如，如果网关能够接受几个压缩算法中的一个，它可以提供一个可接受算法的列表。

下图为本地指令和会话描述符信息流示意图：



4.2.4本地连接的特例

大型网关包括了大量不同类型的端点。在某些网络中，可能经常需要在同一网关内的端点之间建立连接。例如：

- 连接一个呼叫到一个交互式应答设备；
- 连接一个呼叫到一个会议单元；
- 从一个端点到另一个端点的呼叫进行内部选路，经常称为“发卡”连接；

本地连接的建立要比网络连接的建立简单。多数情况下，连接将通过一些本地的互连设备建立，例如，一个 TDM 总线。

当两个端点位于同一网关时，就有可能通过在一个命令中传送要被连接的两个端点名来建立连接。该命令实质上是建立连接（CRCX）命令，只是其中的远端会话描述（Remote session description）

包括了第二个端点的名称。

4.2.5连接的模式

逻辑上，连接组合成“呼叫”（然而，在 MGCP 中，呼叫的概念基本上没有语义上的意义）。多个属于或不属于同一个呼叫的连接可以终止于同一个端点。每个连接用一个“模式”参数来限定，“模式”可以被设置为“只发（sendonly）”、“只收（recvonly）”、“收发（sendrecv）”、“会议（confrnce）”、“非激活（inactive）”、“环回（loopback）”、“一致性检验（conttest）”、“网络环回（netwlopp）”或“网络一致性检验（netwtest）”。

端点上产生的媒体流，将在模式为“只发（sendonly）”、“收发（sendrecv）”或“会议（confrnce）”的连接上发送，而不能在模式为“环回（loopback）”或“一致性检验（conttest）”的连接上发送。然而，在连接上播放信号产生的媒体流始终都可以在连接上发送，与连接的模式无关。

连接上收到的媒体流的处理取决于模式参数：

- 在模式为“只收”、“会议”或“收发”的连接上收到的媒体流被混合并发送给端点，但端点中含有模式为“环回”或“一致性检验”的连接时除外。
- 来自端点的媒体流，可以在模式为“只发”、“会议”或“收发”的连接上传送，但端点中含有模式为“环回”或“一致性检验”的连接时除外。
- 除了发送给端点外，在模式为“会议”的连接上收到的媒体流将会转发给所有模式为“会议”的连接。这也适用于端点中含有一个模式为“环回”或“一致性检验”的连接的情况。转发的细节，例如 RTP 翻译器或混合器，超出了本标准的范围。

注意，为了在连接上检测事件，连接的缺省模式应该是“只收”、“会议”、“收发”、“网络环回”或“网络一致性检测”。事件检测只适用于流入媒体。因此，模式为“只发”、“非激活”、“环回”或“一致性检测”的连接并不检测任何事件，尽管这样的请求并不认为是错误的。

“环回”和“一致性检验”模式用于维护和一致性检验操作。端点或许有一个以上的连接处于“环回”或“一致性检验”模式。只要有一个连接处于这种特殊模式，且端点上又没有其他连接处于不同的维护或检验模式时，维护和检验操作就应该继续进行。有两种一致性检验方法，第一种由 ITU 指定，第二种由 US 使用。第一种方法采用的检测是一个环回测试；发端交换机在承载电路上发送一个信号音（go tone），并期望终端交换机环回该信号音；如果发端交换机收到了同样的环回音，则 COT 一致性检验通过；否则，COT 一致性检验失败。第二种方法，采用的发送和返回信号音不同；发端交换机发送一个信号音（go tone），终端交换机检测到该信号音，并在环回时发送不同的返回音；当发端交换检测到该返回音时，COT 一致性检验通过。否则，COT 一致性检验失败。

如果设为“环回”模式，则期望网关将来自某个端点上的信号从同一个端点环回，该过程的典型应用是，依据 ITU 规范测试中继电路的一致性。如果模式设为“一致性检验”，则网关被通知电路的另一端已经依据 GR 规范发启一致性检验，网关将按照双音一致性检验的要求把电路置为应答模式。

如果设为“网络环回”模式，则来自连接的音频信号将在同一个连接上回送过去。媒体流并不转发给端点。

如果设为“网络一致性检验”模式，网关将按照双音一致性检验要求的应答模式来处理连接上接收的包，并将处理过的信号发回到该连接上。媒体流并不转发给端点。规定“网络一致性检验”模式只是考虑到后向兼容，并不主张使用它。

4.2.6呼叫代理及其它实体的命名

MGCP 协议被设计成允许实现冗余的呼叫代理，以提高网络可靠性。这意味在实体和硬件平台或网络接口之间没有固定的绑定关系。

与端点名称类似，呼叫代理名称也包括两部分。从语法上来说，名称的本地部分并不展示任何内部结构。例如，呼叫代理名称可以是：

```
ca1@ca.whatever.net
```

需要注意，本地名和域名都必须提供。但也主张在实施中接受只包含域名的呼叫代理名。

可靠性可以通过以下过程来增强：

- 实体（比如端点）或呼叫代理用域名标识，而不用它们的网络地址标识。一个域名可以和多个网络地址相关，如果一个命令或响应无法传送到其中某个地址，实现上必须尝试用另一个地址进行传送。
- 实体可以转移到另一个平台。逻辑名（域名）和实际平台之间的关联保存在 DNS 服务器里，呼叫代理和网关必须追踪 DNS 服务器中有关它们记录，并通过查询 DNS 服务器来刷新已经过期的信息。

除了使用域名和 DNS 服务器提供间接可靠性支持外，通知实体（notified entity）的概念是本协议关于可靠性及错误倒换（failover）的核心。一个端点的通知实体是当前正在控制该端点的呼叫代理。在任何时候，一个端点有且只有一个通知实体和它相关联。通知实体决定了端点将命令发送到哪里；当端点需要向呼叫代理发送一条命令时，它必须发命令给当前的通知实体。然而，通知实体并不决定命令从哪里收到；任何呼叫代理都可以向端点发送命令。请参考本规范关于安全部分的描述。

在启动时，通知实体（notifiedentity）必须被设定为预定值。呼叫代理发出的多数命令有能力通过使用“ NotifiedEntity ”参数来明确指定通知实体。除非收到一个新的通知实体参数或者端点重新启动（冷启动或热启动）外，通知实体将保持不变。

如果一个发送的“ NotifiedEntity ”参数是“ 空 ”值，则端点的通知实体将被设为空值。如果一个端点的通知实体为空值或没有明确设定（通过命令或预置），被通知实体将被默认为该端点收到的最后一个成功的非审查命令的源地址（IP 地址和 UDP 端口号）。审查不改变通知实体。本标准不主张使用空的“ NotifiedEntity ”参数值，因为它容易导致错误，且会消除基于 DNS 的错误倒换和可靠性机制。

4.3 号码表

呼叫代理可以要求网关收集用户拨打的号码，这种机制主要用于住宅网关收集用户的拨号，也可以用于中继网关或接入网关收集接入码、信用卡号码和其它呼叫控制业务请求的号码。

网关在每收到一位拨号时都立即通知呼叫代理是一种有效的收号方法，但这种方法会产生大量的交互信息。比较可取的方法是，网关先把拨号信息保存在缓存中，然后再用一条消息把它们传送出去。

但是采用缓存保存的问题是，网关很难预测发送前要收集多少位号码。例如，使用桌上的电话，我们可以拨打如下号码：

0	本地话务员
00	长途话务员
xxxx	本地扩展号码
8xxxxxxx	本地号码
#xxxxxxx	在其它场所的本地号码快捷形式
*xx	星号业务
91xxxxxxxxxx	长途号码
9011 + 最多 15 个数字	国际长途号码

解决这个问题的方法是由呼叫代理给网关加载一个符合拨号方案的号码表（DigitMap）。号码表的表示方法源自 UNIX 系统命令 egrep 的语法。例如，上述的拨号计划可以描述成下面的号码表：

```
(0T| 00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T)
```

号码表的正式语法在本协议的正式语法描述的 DigitMap 规则部分定义（见附录 A），支持基本的号码表字母是必需的，而支持扩展的号码表字母是可选的。网关收到带有扩展号码表字母的号码表而不支持时，应该返回错误码 537（unknown digit map extension）。

根据该语法生成的号码表由一个字符串或多个字符串列表（字符大小写无关）来定义。列表中的每一个字符串都是一个可选的编号方案，它可以表示成一组数字或定时器，也可以表示成网关能找到最短可能匹配的表达式。以下成分可以被用于每个编号方案中：

- 数字：数字“ 0 ”到“ 9 ”。
- 定时器：符号“ T ”表示一个定时器，用于匹配定时器超时事件。
- DTMF：一个数字或一个定时器，或符号“ A ”、 “ B ”、 “ C ”、 “ D ”、 “ # ”、 “ * ”中的一个。还可以定义扩展符号。
- 通配符：符号“ x ”代表任何数字（“ 0 ”到“ 9 ”）。
- 范围：包括在方括号（“ [”和“] ”）中的一个或多个 DTMF 符号。
- 子范围：由 “ - ”分开的两个数字，代表这两个数字（包括它们）之间的数字。子范围只能应用在范围内，即包含在“ [”和“] ”中。
- 位置：一个点号“ . ”表示对它前面成分的任意多次（包括零次）的匹配。

网关按照号码表检测匹配事件的操作如下：

1) 把事件代码作为标记加到内部状态变量“ 当前拨号字符串“ （current dial string）的末尾。
2) 把当前拨号字符串和号码表比较，尝试匹配号码表中的每一个条目。
3) 如果匹配结果是没有确定（部分匹配号码表中至少一个条目，且不能完全匹配另外的条目），则没有进一步的动作。

如果结果匹配了一个条目，或严格匹配（over-qualified，也就是不可能进一步匹配任何数字），则把当前收集的事件序列发送给呼叫代理。在本规范中提到的匹配可以是两种情况：“ 完全匹配 ”，指恰好匹配号码表中的一个条目；或“ 不匹配 ”，即所拨号码不匹配号码表中任何条目，例如，意外的定时器事件，就能引起“ 不匹配 ”。“ 完全匹配 ”和“ 不完全匹配 ”均能触发对所收集数字（也可能包括其他事件）的上报。

以下用例子对上述内容进行说明：

假定号码表为（xxxxxxx|x11），且当前拨号串是“ 41 ”，如果输入“ 1 ”，当前拨号串变为“ 411 ”，那么就有一个对“ xxxxxxx ”的部分匹配，但有一个对“ x11 ”的完全匹配，因此向呼叫代理发送“ 411 ”。

以下的号码表例子更加精妙：

假定号码表为（0[12].|00|1[12].1|2x.#）。

如果输入“ 0 ”，会立即对“ 0[12]. ”产生一个匹配（因为“ . ”号允许前面的[12]发生次数为零），因此，该号码表无法产生“ 00 ”这样的输入。

如果输入“ 1 ”，仅会对“ 1[12].1 ”产生一个部分匹配。输入“ 12 ”也仅仅是一个部分匹配。但是，“ 11 ”和“ 121 ”都是一个完全匹配。

如果输入“ 2 ”，会产生一个部分匹配，输入“ 23 ”、“ 234 ”、“ 2345 ”等仍然是产生部分匹配。如果不进一步输入“ # ”号，就不会产生完全匹配。譬如，输入“ 2# ”、“ 2345# ”就会产生匹配。

注意，号码表仅定义了一种根据语法匹配事件编码序列的模式。尽管这里定义的号码表用于 DTMF 输入，但也可以通过定义扩展包使号码表用于其它类型事件代码的输入，只要这些类型的事件代码遵从为 DTMF 事件（如数字、“ T ”）定义的语法。在可以想象到的使用场合，特定事件的定义都应该在包定义中显式说明。

由于号码表没有容量限制，建议网关对每个端点支持的最大号码表容量为至少 2048 字节。

4.4 包

MGCP 是一个模块化且可扩展的协议，然而伴随可扩展性，也带来了对每一个扩展的管理、标识及命名的问题。这个问题通过包的概念得到解决，包是简单又很明确的扩展组。例如，一个包可能支持某一组有关模拟接入线路的事件和信号（如摘机和振铃），另一个包可能支持另一组有关模拟接入线路或其它类型的端点（如视频）的事件和信号。一个端点可以支持一个或多个包。

MGCP 在一个包中允许定义以下类型的扩展：

- 承载信息（BearerInformation）
- 本地连接选项（LocalConnectionOptions）
- 扩展参数（ExtensionParameters）
- 连接模式（ConnectionModes）
- 事件（Events）
- 信号（Signals）
- 动作（Actions）
- 号码表字母（DigitMapLetters）
- 连接参数（ConnectionParameters）
- 启动方法（RestartMethods）

- 原因码（ReasonCodes）
- 响应码（Return codes）

上述每一类型都会在后面详细解释。在一个包中定义这些扩展项的规则在第 11 章中规定，相应的编码和语法分别定义在第 5 章及附录 A 中。

除了号码表字母（DigitMapLetters）外，包通过使用包名作为扩展项前缀的命名方法为每类扩展项定义一个独立的名字空间，即：package-name/extension ，也就是包名后面紧跟 “ / ”和扩展项名。

一个端点如果支持一个或多个包可以定义其中一个作为该端点的缺省包，使用缺省包中的事件和信号时可以不加包名，但建议也带包名，其它扩展项除了号码表字母（DigitMapLetters）外，在引用时都必须加包名。

包名是与大小写无关的数字、字母及短线组成的字符串，但短线不能是第一或最后一个字符。例如，包名可以是“ D ”、“ T ”及“ XYZ ”。包名对大小写不敏感，因此，“ XYZ ”、 “ xyz ”和“ xYz ”是相同的包名。

包的定义由其它文件提供，包名和扩展项名必须在 IANA 注册。包的定义规则详见第 11 章。

实施者可以通过使用实验包来获取经验，实验包的名字必须以“ x- ”打头，IANA 不应该注册以这两个字母开头的包名，也不应注册以“ x+ ”开头的包名，“ x+ ”也是保留字母。网关收到含有它所不支持的包的命令时，必须返回一个错误（推荐使用错误码 518 - unsupported package）。

4.5 事件和信号

事件和信号的概念是 MGCP 协议的核心。呼叫代理可以通过使用请求事件（RequestedEvents）参数中的事件名来要求端点通知其上发生的某事件（比如摘机事件）。呼叫代理可以通过使用信号请求（SignalRequests）参数中的事件名来向端点加载某个信号（比如拨号音）。

事件和信号被打成包，在包内它们共享同一个名字空间，即以下称作事件名。事件名是与大小写无关的字母、数字、及短线组成的字符串，且短线不能是第一或最后一个字符。一些事件代码或许需要其他参数数据，这可以通过在一对圆括号间增加参数来实现。事件名是不区分大小写的，比如“ hu ”、 “ Hu ”、“ HU ”或“ hU ”都是一样的。

例如，事件名可以是“ hu ”（摘机），“ hf ”（拍叉簧）或“ 0 ”（数字 0）。

端点的缺省包中的事件可以不带包名，但建议带上包名。如果事件名不带包名，则必须使用该端点缺省的包名。例如，对于模拟接入线路，它有一个缺省包“ L ”，L 包中定义了一个事件“ dl ”（拨号音），则 L/dl 和 dl 两个事件名是等同的。

而对于这个端点上的任何非缺省包，例如，模拟接入端点上的通用包（G），事件名必须带上包名。再次强调，建议无条件地带上包名。

在某些包中支持数字或字母，特别是“ DTMF ”包，定义了符合号码表规则的数字和字母。号码表规则定义了数字（0 到 9）， “ * ”， “ # ”，字母（“ A ”、“ B ”、“ C ”、“ D ”）以及定时器指示符“ T ”。这些数字和字母可以组合成字符串用于代表用户拨打的按键，此外，字母“ X ”可以用于表示任何数字（0 到 9）。并且，在扩展时还可以定义其它字母。本协议为了易于表示拨号字符串，而对事件的命名形式做出以下规定：一个非拨号的事件名必须至少含有一个既不是数字，也不是字母 A、B、C、D、T 或 X 的字符，事件名也不能只包含一个特殊符号“ * ”或“ # ”。但是，由超

过一个字符组成的事件名可以使用上述任何字符。

呼叫代理或许经常必须要求网关检测一组事件。表示这些事件组有两个规定：

- 通配符“ * ”和“ all ”可用于检测某个包的任何事件，或者在多个包中的指定事件，或者网关所支持的任何包中的任何事件。
- 规则的范围表示符（Range notation）可以用于检测一定范围内的数字。
- “ * ”通配符可以用来代替包名，关键字"all"可以作为通配符来代替事件名：
- 类似“ foo/all ”的名字表示包“ foo ”内的所有事件。
- 类似“ */bar ”的名字表示网关支持的任何包中的事件“ bar ”。
- 名字“ */all ”表示端点支持的全部事件。

本标准建议不对“ 所有包 ”和“ 所有事件 ”通配符定义其它的细节。因为，那样做带来的好处有限，却引入了很多复杂性和潜在错误。因此，并不建议使用它们。

呼叫代理或许要求网关检测一组数字或字母，这些数字字母或者单独描述，或者使用数字串语法中定义的范围（Range）符号。例如，呼叫代理能够：

- 用字母“ x ”表示任何从 0 到 9 的数字；
- 用记号“ [0-9#] ”表示数字 0 到 9 和“ # ”号；

单独的事件代码仍然在包中定义（例如，“ DTMF ”包）。

缺省地，事件只能在端点上产生和检测，然而事件也能够定义在连接上产生和检测。例如，要求网关在连接上提供回铃音。当事件要对连接提供时，连接名要紧跟在事件名之后，使用一个“ @ ”符号作分隔符，比如：G/rt@0A3F58，这里，“ G ”是包名，“ rt ”是事件名。当正在产生或检测事件的连接被删除时，要停止其上的事件检测和信号产生。这还取决于不同信号，有可能产生一个错误。

通配符“ * ”可以用来表示“ 所有连接 ”。此时，网关将在连接了指定端点的所有连接上产生或检测事件，这适用于现存的和将要在该端点建立的所有连接。例如：R/qa@*。这里，“ R ”是包名，“ qa ”是事件名。

当处理一个使用“ 所有连接 ”（*）通配符的命令时，该通配符适用于端点上所有现存的和将来的连接，“ * ”不会被展开，如果后来的命令显式或隐式地涉及到该事件，仍将使用该“ * ”值。只是当实际检测到事件时，该事件要包含产生它的特定连接的名称。

通配符“ $ ”表示“ 当前连接 ”，它只能由呼叫代理来使用，且用在事件通知请求被封装在建立或修改连接命令中的情况下，此时，网关将在当前正在建立或修改的连接上产生或检测事件。例如，G/rt@$。

当处理一个带有“ 当前连接 ”通配符（$）的命令时，该通配符将被展开为当前连接的值，如果后来的命令显式或隐式地涉及到该事件，都将使用其展开值。也就是说，“ 当前连接 ”通配符（$）只被展开一次，并且是在显式包含它的命令最初被处理的时候。

连接标识符（或由通配符替代）可以结合“ 所有包 ”和“ 所有事件 ”约定来使用。例如，“ */all@* ”表示端点上所有当前与将来连接的所有事件。然而，就像前面说过的，本标准极不主张使用“ 所有

包 ” 和 “ 所有事件 ” 通配符。

信号依据其行为的不同，分为几种类型：

- on/off（OO）：通断信号，一旦使用，信号就一直持续到被关掉为止。只有在重新启动或收到新信号请求（SignalRequests）参数时发生显式的关闭信号。OO 类型信号被定义为等幂的（idempotent），因此，对它的多次开或关的请求都是有效的且不会出错。OO 类型信号可能是可视的消息等待指示信号（VMWI），一旦打开，除了被来自呼叫代理的明确命令关闭或因端点重启动关闭外，这种信号不能被关闭。检测到的事件也不能使这类信号关闭。

- Time-out（TO）：超时信号,一旦使用，信号一直持续到被取消（通过事件的发生或后来的信号列表（可能为空）中没有包含该信号）或信号定时器超时。在 TO 信号超时时会产生一个“ 操作完成 ”（operation complete）事件。“ 回铃音 ”（Ringback）是一个在 180 秒后超时的 TO 信号,如果一个事件在 180 秒前发生,缺省行为是停止该信号（“ 保持信号激活（Keep signals active） ” 动作会覆盖该行为）。如果信号不被停止，它会超时而停止并产生一个 “ 操作完成 ”（Operation complete）事件，该事件可能被呼叫代理请求，也可能不请求。如果呼叫代理请求通知 “ 操作完成 “ （Operation complete）事件，则发往呼叫代理的该事件应该包含超时的信号名（注意，如果信号还有参数，则不需要报告参数）。如果信号由某一连接产生，还应在事件的通知中包含连接名。TO 信号可定义缺省的超时值，该缺省值可以被预设置改变，超时值作为信号（signals）的参数。超时值为零意味着没有超时限制。如果一个已经启动的 TO 信号在产生 “ 操作完成 ”（Operation complete）事件之前播放失败，那么会产生一个 “ 操作失败 ”（Operation failure）事件，其中包含产生错误的信号名，删除一个带有激活的 TO 信号的连接就会导致这种错误的产生。

- Brief（BR）：简短信号,信号的持续时间非常短并可以自行停止。如果一个信号停止事件发生，或者加载了一个新的信号请求，当前激活的 BR 信号将不会停止，然而，任何悬置还未发送的 BR 信号都要被取消（如果 NotificationRequest 参数中包含 BR 信号，而当前已经有了一个激活的 BR 信号，则 NotificationRequest 参数中的 BR 信号变为悬置）。例如，一个简短音可能是一个 DTMF 数字，如果 DTMF 数字 “ 1 ” 正在播放，而信号停止事件发生了，则 “ 1 ” 将会完成播放；如果在 “ 1 ” 完成播放前，收到了播放 DTMF 数字 “ 2 ” 的请求，则数字 “ 2 ” 将变为悬置信号。

连接上产生的信号必须包含连接名。

5 命令与参数

5.1 一般描述

MGCP 协议通过一系列事务来实现媒体控制接口。事务由命令及必要的响应组成。命令的作用包括连接处理和端点处理。

所有命令都由命令头以及随后任选的会话描述组成。所有响应都由响应头以及随后任选的会话描述组成。MGCP 使用事务标识符来关联命令和响应。事务标识符编码为命令头的一部分，同时作为响应头的一部分重复出现。

头部和会话描述编码为一组文本行，用回车和换行符（或单个的换行符）分开。会话描述由空行开始。

注意，MGCP 的 ABNF 语法描述包含在附录 A 中。命令和响应应该根据语法来编码，除 SDP 外都是与大小写无关的。类似的，实现中应该能够解析符合该语法规则的命令和响应。此外，推荐具体的实现方案容忍一行中多余的空白符。

一些产品允许使用引用字符串，它有必要用于避免语法问题。使用引用字符串形式时，其内容应该采用 UTF-8 编码，且提供的实际值是非引用字符串（UTF-8 编码）。在使用引用字符串或非引用字符串两种形式时，只要不违背语法规则，使用其中任何形式都可以。

5.1.1事务标识符

MGCP 使用事务标识符来关联命令和响应。事务标识符编码为命令头的一部分，同时作为响应头的一部分重复出现。网关支持两种独立的事务标识符名字空间：

- 用于发送事务的名字空间；
- 用于接收事务的名字空间；

至少，在控制网关的所有呼叫代理中，发给一个特定网关的命令的事务标识符在事务的最大生存周期内必须是唯一的。因此，不考虑发送命令的呼叫代理，网关通过简单地检查事务标识符就可以发现重复的事务。然而，呼叫代理之间的事务标识符的协调不在本标准讨论范围之内。

从一个特定的网关发出所有命令的事务标识符在事务的最大生存周期内必须是唯一的，而不管命令发送到哪一个呼叫代理。因此，呼叫代理能够通过结合端点域名和事务标识符来检测来自网关的重复事务。

事务标识符的编码为十进制数字组成的字符串，最多 9 个字符。在命令行里，该字符串紧跟在命令动词编码后面。

事务标识符的值可以取 1 到 999999999（包括 1 和 999999999）之间的任何值。事务标识符不应该以零开头，尽管有没有零在数值上是相等的（开头的零被忽略）。一个 MGCP 实体在执行完前一个命令后，3 分钟内不得再次使用该命令用过的事务标识符。

端点标识符和实体名称编码采用与大小写无关的 E-Mail 地址方式（在 RFC 821 中定义），但对名称的本地部分有一些语法上的限制。进一步说，本地端点名和域名部分均最多为 255 个字符。在这些地址中，域名部分标识了端点所属的系统，左边部分则标识了系统上的特定端点或实体。

地址的例子如下：

hrd4/56@gw23.example.net	在“Example”网络的网关 23 中的“hrd4”接口内的第 56 条电路
Call-agent@ca.example.net	“example”网络的呼叫代理
Busy-signal@ann12.example.net	录音通知服务器 12 中的“busy signal”虚端点。

通知实体名的语法采用和端点标识符相同的语法来表示，只是有可能增加一个端口号，比如：Call-agent@ca.example.net:5234

通知实体名中省略了端口号的情况下，必须使用缺省的 MGCP 呼叫代理端口号（2727）。

5.1.2命令的组成

所有命令都由命令头以及随后任选的会话描述组成。命令头由以下两部分组成：

- 一个命令行，指明请求的命令动词、事务标识符、动作所请求的目标端点以及 MGCP 协议版本号。
- 零或多个参数行，由参数名及随后的参数值构成。

除非其他参考标准（例如，SDP）另行规定，命令头中的每一部分对都与大小写无关。这适用于命令动词、参数及参数值。因此，所有对大小写以及它们的任意组合的比较结果都是相等的。

5.1.2.1 命令行

命令行的组成包括：

- 被请求的命令动词
- 事务标识符
- 执行此命令的端点名称（在通知或重启命令中，指发布此命令的端点名）
- 版本号

这四部分均编码为可打印的 ASCII 字符串，用空白符分开，空白符为 ASCII 空格符（0x20）或制表符（0x09）。推荐只使用单个 ASCII 空白符来分隔。然而，MGCP 实体必须能够解析带有多余的空白符的消息。

5.1.2.2 动词的编码

被请求的命令动词由 4 个大写或小写的 ASCII 码组成（与大小写无关），定义见下表：

命令	中文名	命令动词
EndpointConfiguration	端点配置	EPCF
CreateConnection	建立连接	CRCX
ModifyConnection	修改连接	MDCX
DeleteConnection	删除连接	DLCX
NotificationRequest	通知请求	RQNT
Notify	通知	NTFY
AuditEndpoint	审查端点	AUEP
AuditConnection	审查连接	AUCX
RestartInProgress	正在重启动	RSIP

在以后的协议版本中可能定义新的命令。有时，为了试验目的，需要在新协议出版前使用未经批准的新命令，称为试验命令。试验命令的编码也为四个字符，但以 X 开头，例如，XPER。

5.1.2.3 协议版本的编码

协议版本编码为关键字 MGCP，加上紧随其后的空白符和版本号，还可能包括一个简档（profile）名。版本号包含一个十进制编码的主版本号，一个逗点，一个十进制编码的小版本号。本协议的版本号是 1.0。

如果具有简档（profile）名，它会用可视字符串来表示，位于行的末尾，并以空白符与其前面的内容分隔。简档（profile）名或许为想要对 MGCP 协议实行限制或其他描述（Profiling）的用户定义。

在初始消息中，版本号编码为： MGCP 1.0

一个实体在收到带有它所不支持的版本号的命令时，必须响应一个错误（推荐使用错误码 528 - incompatible protocol version）。注意，这也适用于不支持简档（profile）名时。

5.1.2.4 参数行

参数行由参数名（多数情况下由一到两个字符组成）、冒号、空格符（可选）和参数值组成。命令里出现的参数定义见下表：

参数名	中文名	编码	参数值
BearerInformation	承载信息	B	参见参数描述。
CallId	呼叫标识符	C	参见参数描述。
Capabilities	性能	A	参见参数描述。
ConnectionId	连接标识符	I	参见参数描述。
Connection Mode	连接方式	M	参见参数描述。
ConnectionParameters	连接参数	P	参见参数描述。
DetectEvents	检测事件	T	参见参数描述。
DigitMap	数图	D	数图的文本编码。
EventStates	事件状态	ES	参见参数描述。
LocalConnectionOptions	本地连接选项	L	参见参数描述。
MaxMGCPDatagram	最大 MGCP 数据报	MD	参见参数描述。
NotifiedEntity	通知实体	N	一个符合 RFC821 格式的标识符，由任意一个字符串和请求实体的域名组成，也可能在末尾带有一个端口号。比如，Call-agent@ca.example.net:5234。参见参数描述。
ObservedEvents	观察事件	O	参见参数描述。
PackageList	包列表	PL	参见参数描述。
QuarantineHandling	隔离处理	Q	参见参数描述。
ReasonCode	原因码	E	一个含有三个数字的字符串,有可能后随任意字符串。参见参数描述。
RequestedEvents	请求事件	R	参见参数描述。
RequestedInfo	请求信息	F	参见参数描述。
RequestIdentifier	请求标识符	X	参见参数描述。
ResponseAck	响应确认	K	参见参数描述。
RestartDelay	重启延迟	RD	十进制编码表示的秒数。
RestartMethod	重启方法	RM	参见参数描述。
SecondConnectionId	第二连接标识符	I2	Connection Id.
SecondEndpointID	第二端点标识符	Z2	Endpoint Id.
SignalRequests	信号请求	S	参见参数描述。
SpecificEndpointID	指定端点标识符	Z	一个符合 RFC821 格式的标识符，由任意一个字符串，后随“@”和端点所属的网关的域名组成。参见参数描述。

参数名	中文名	编码	参数值
RemoteConnection Descriptor	远端连接描述符	RC	参见参数描述。
LocalConnection Descriptor	本地连接描述符	LC	参见参数描述。

这些参数不必要在所有的命令中都出现。下表给出了参数和命令之间的关系，M 代表必选参数，O 代表任选参数，F 代表禁止使用。除非另行指定，一个参数不允许出现多于一次。

Parameter name （参数名）	EP CF	CR CX	MD CX	DL CX	RQ NT	NT FY	AU EP	AU CX	RS IP
BearerInformation	O*	O	O	O	O	F	F	F	F
CallId	F	M	M	O	F	F	F	F	F
Capabilities	F	F	F	F	F	F	F	F	F
ConnectionId	F	F	M	O	F	F	F	M	F
ConnectionMode	F	M	O	F	F	F	F	F	F
Connection-Parameters	F	F	F	O*	F	F	F	F	F
DetectEvents	F	O	O	O	O	F	F	F	F
DigitMap	F	O	O	O	O	F	F	F	F
EventStates	F	F	F	F	F	F	F	F	F
LocalConnection-Options	F	O	O	F	F	F	F	F	F
MaxMGCPDatagram	F	F	F	F	F	F	F	F	F
NotifiedEntity	F	O	O	O	O	O	F	F	F
ObservedEvents	F	F	F	F	F	M	F	F	F
PackageList	F	F	F	F	F	F	F	F	F
QuarantineHandling	F	O	O	O	O	F	F	F	F
ReasonCode	F	F	F	O	F	F	F	F	O
RequestedEvents	F	O	O	O	O*	F	F	F	F
RequestIdentifier	F	O*	O*	O*	M	M	F	F	F
RequestedInfo	F	F	F	F	F	F	O	M	F
ResponseAck	O	O	O	O	O	O	O	O	O
RestartDelay	F	F	F	F	F	F	F	F	O
RestartMethod	F	F	F	F	F	F	F	F	M
SecondConnectionId	F	F	F	F	F	F	F	F	F
SecondEndpointId	F	O	F	F	F	F	F	F	F
SignalRequests	F	O	O	O	O*	F	F	F	F
SpecificEndpointId	F	F	F	F	F	F	F	F	F
RemoteConnection-Descriptor	F	O	O	F	F	F	F	F	F
LocalConnection-Descriptor	F	F	F	F	F	F	F	F	F

注释（*）：

* 承载信息参数仅在一定条件下是任选参数。参见参数描述。

* 请求标识符在生成连接、修改连接和删除连接命令中是任选参数，但是当以上命令包含封装的

通知请求时，该参数为必选参数。

* 请求事件参数和信号请求参数在通知请求命令中是任选参数。如果这些参数省略，则认为相应的列表是空的。
* 连接参数仅在由网关发送的删除连接请求中有效。

参数集可以采用以下两种不同的方式来扩展：

- 包扩展参数（推荐）
- 供应商扩展参数

包扩展参数在包中定义，为推荐的扩展方式，有以下优势：

- 包名有注册机制（IANA）
- 参数有独立的名空间
- 扩展能方便地分组
- 具有通过审查决定是否支持的简单方式

如果实施中（例如，开发新的 MGCP 应用）需要试验新参数，可以使用供应商参数。供应商参数必须以字符串“ X- ”或“ X+ ”开头的名字来标识。例如，

```
X-Flower: Daisy
```

参数名以“ X+ ”开头表示必选参数扩展。MGCP 实体收到了一个不能识别的必选参数扩展时，它应该拒绝执行命令且返回错误代码 511（unrecognized extension）。

参数名以“ X- ”开头表示可选参数扩展，MGCP 实体收到不能识别的该类参数后，必须忽略该参数。

注意，供应商扩展参数使用不可管理的名字空间，会有潜在的名字冲突。因此，鼓励供应商在他们的扩展参数中包含他们的特定字符串。例如，供应商名称。

5.1.3响应的组成

所有响应都由响应头以及随后任选的会话描述组成。响应头由以下部分组成：

- 一个响应行，指明对请求的动作的响应；
- 可选的响应参数行；

以下是响应头的例子： 200 1203 OK

5.1.3.1 响应行

响应行以响应码开头，响应码用三位数字表示，后跟空白符、事务标识符。在包中定义的响应码（8xx）后跟空格，斜线（“ / ”）以及包名。此外，所有的响应码都可以后跟以空白符开头的可选的注释信息。

5.1.3.2 参数行

下表描述了在响应头中出现的强制和任选参数的情况，作为触发响应的命令的功能。M 代表必备的，O 代表任选的，F 代表禁止的。除非另行规定，一个参数不允许出现多于一次。注意，下表仅

反映没有定义其它行为的响应的缺省行为。如果收到的响应带有不理解或禁止的参数，这些参数必须被简单地忽略。

Parameter name （参数名）	EPCF	CRCX	MDCX	DLCX	RQNT	NTFY	AUEP	AUCX	RSIP
BearerInformation	F	F	F	F	F	F	O	F	F
CallId	F	F	F	F	F	F	F	O	F
Capabilities	F	F	F	F	F	F	O*	F	F
ConnectionId	F	O*	F	F	F	F	O*	F	F
ConnectionMode	F	F	F	F	F	F	F	O	F
Connection-Parameters	F	F	F	O*	F	F	F	O	F
DetectEvents	F	F	F	F	F	F	O	F	F
DigitMap	F	F	F	F	F	F	O	F	F
EventStates	F	F	F	F	F	F	O	F	F
LocalConnection-Options	F	F	F	F	F	F	F	O	F
MaxMGCPDatagram	F	F	F	F	F	F	O	F	F
NotifiedEntity	F	F	F	F	F	F	O	O	O
ObservedEvents	F	F	F	F	F	F	O	F	F
QuarantineHandling	F	F	F	F	F	F	O	F	F
PackageList	O*	O*	O*	O*	O*	O*	O	O*	O*
ReasonCode	F	F	F	F	F	F	O	F	F
RequestIdentifier	F	F	F	F	F	F	O	F	F
ResponseAck	O*	O*	O*	O*	O*	O*	O*	O*	O*
RestartDelay	F	F	F	F	F	F	O	F	F
RestartMethod	F	F	F	F	F	F	O	F	F
RequestedEvents	F	F	F	F	F	F	O	F	F
RequestedInfo	F	F	F	F	F	F	F	F	F
SecondConnectionId	F	O	F	F	F	F	F	F	F
SecondEndpointId	F	O	F	F	F	F	F	F	F
SignalRequests	F	F	F	F	F	F	O	F	F
SpecificEndpointId	F	O	F	F	F	F	O*	F	F
LocalConnection-Descriptor	F	O*	O	F	F	F	F	O*	F
RemoteConnection-Descriptor	F	F	F	F	F	F	F	O*	F

注释（*）：

* 包列表（PackageList）参数仅适用于响应码 518（unsupported package），但审查端点（AuditEndpoint）命令是个例外，该参数在被审查时也可能被返回。
* 响应证实（ResponseAck）参数，适用于正在处理的事务在临时响应后发送的最终响应，不可用于其他响应。在这种情况下，出现响应证实参数应该触发一个响应证实---任何提供的响应证实参数值都将被忽略。
* 在建立连接消息的情况下，响应行后随一个连接标识符（Connection-Id）参数和一个本地连接描述符（LocalConnectionDescriptor）参数。如果建立连接请求发送到使用通配符的端点标识

符，响应行也可以后随一个特定端点标识符（SpecificEndpointID）参数。连接标识符（Connection-Id）参数和本地连接描述符（LocalConnectionDescriptor）参数在表中被标示为可选参数。事实上，在建立一个连接时，它们在所有的肯定响应中是必选参数；在失败响应中，它们是被禁止使用的，此时，没有连接被建立。

* 本地连接描述符（LocalConnectionDescriptor）参数必须在对建立连接（CRCX）命令的肯定响应（响应代码 200）中传送。该参数也必须在对导致会话参数修改的修改连接（MDCX）命令的响应中传送。本地连接描述符（LocalConnectionDescriptor）参数采用“会话描述”编码，并用空行与响应头分隔。
* 连接参数（Connection-Parameters）仅在对未使用通配符的删除连接（DLCX）命令的响应中有效，该命令由连接代理发送。
* 多个连接标识符（ConnectionId）、特定端点标识符（SpecificEndpointId）及性能（Capabilities）参数可以包含在对审查端点（AUEP）命令的响应中。
* 当几个会话描述符包含在同一个响应中时，它们一个接一个来放置，用空行分隔。例如，在对一个审查连接请求的响应中，同时包含本地会话描述符和远端会话描述符，就会发生这种情况：

```
200 1203 OK
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:PCMU;G726-32
M: sendrecv
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27,LA=48

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 1296 RTP/AVP 0

v=0
o=- 33343 346463 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

在该例中，根据 SDP 语法，每一个描述符以“版本”行（v=…）开头。本地描述符始终在远端描述符前传送。如果请求的连接描述符中的连接并不存在，则连接描述符仅具有“版本”行（v=…）。

5.2 MGCP参数

5.2.1BearerInformation（承载信息）

承载信息值采用逗号分隔的属性序列来编码。属性由属性名，以及其后可能的冒号与属性值来

表示。

唯一定义的属性是编码属性（编码“ e ”），该属性必须取值为“ A ”律（A-law）或“ mu ”律（mu-law）。

承载信息编码的例子：B: e:mu

承载信息的属性集可以通过包来扩展。

5.2.2 CallId （呼叫标识符）

呼叫标识符编码为十六进制字符串，最大长度 32 个字符。呼叫标识符作为字符串而不是数值来比较。

5.2.3Capabilities （性能）

当审查时，性能参数通知呼叫代理有关端点的性能。对性能和本地连接选项中共同的参数，性能编码基于本地连接选项编码，但仍有一个不同的参数行编码“ A ”。另外，性能还包括支持的信息包序列和支持的方式序列。

性能中使用的参数是：

- 一个所支持的编解码序列。后面的参数将应用此序列中指定的所有的编解码。如果需要指定一些参数（比如静音抑制）只和某些编解码兼容，那么网关将返回多个性能参数；每个参数对应一组编解码。
- 打包周期：可以指定一个范围。
- 带宽：可以指定一个和打包周期范围相适应的范围（假设没有静音抑制）。如果缺省，该值将从编解码类型导出。
- 回声抑制：如果支持设为“ on ”，否则为“ off ”，默认值为支持。
- 静音抑制：如果支持设为“ on ”，否则为“ off ”，默认值为支持。
- 增益控制：值“ 0 ”表示不支持增益控制，所有其它值表示支持，默认值是支持。
- 业务类型：值“ 0 ”表示不支持业务类型，其它值表示支持，默认值为支持。
- 资源预留业务：本参数表明除了“ 尽最大努力 ”外所支持的预留业务。值“ g ”表示网关同时支持有保证业务和负载控制业务；“ cl ”表时只支持负载控制业务。默认值为“ 尽最大努力 ”。
- 密钥：任何编码值都表示支持加密，默认是不支持加密（通过省略该参数来暗指）。
- 网络类型：关键字“ nt ”，后跟冒号以及用分号分隔的支持的网络类型。该参数是任选参数。
- 包：端点支持的包被编码为关键字“ v ”，后跟冒号和字符串。如果指定了一个值序列，值间用分号隔开。序列中第一个包是端点默认的包。
- 方式：端点支持的方式被编码为“ m ”，后跟冒号以及用分号分隔的此端点支持的连接方式序列。

缺乏对某一性能的支持也可以通过在性能集中不包含该参数来指明。

性能的例子如下：

```
A: a:PCMU;G728, p:10-100, e:on, s:off, t:1, v:L,
                    m:sendonly;recvonly;sendrecv;inactive
```

该例中的换行仅仅是出于格式化的原因，实际实施中是不允许的。

如果需要返回多个性能，则每一个都要作为一个独立的性能行来返回。

由于本地连接选项能被扩展，因而性能参数序列也能被扩展。每个扩展项可以定义它们如何作为性能来报告。如果没有这样的定义，使用以下缺省规则：

- 包扩展属性：并不报告单个属性。只要简单地在支持的包序列中报告包名。
- 供应商扩展属性：仅报告属性名，不带值。
- 其它扩展属性：仅报告属性名，不带值。

5.2.4 Coding of Event Names （事件名编码）

事件名由一个任选的包名，用斜杠（“ / ”）与实际事件名分开来组成。通配符星号“ * ”可以用来表示所有包。事件名后可跟任选的 AT 符号（@）以及观察到事件的连接标识符。事件名用于请求事件（RequestedEvents）、信号请求（SignalRequests）、观察事件（ObservedEvents）、检测事件（DetectEvents）以及事件状态（EventStates）参数中。

事件和信号可以通过其上定义的参数来限定。这些参数可以放在双引号中（事实上，由于语法限制，一些参数必须放在双引号中），采用 UTF-8 编码。

对事件和信号来说，参数名“ ！ ”（叹号）都保留为将来使用。

每一个信号都有一个如下的信号类型和它相关：通断信号(OO)，超时信号(TO)，简短信号(BR)（这些信号类型在包定义中已经描述，在此处不再介绍）。通断信号可以用“ + ”号打开，用“ - ”号关闭，如果没有指定，则默认为打开。下面的两种情况均可以打开信号 vmwi：

```
L/vmwi(+)

  L/vmwi
```

除了“ ！ ”、“ + ”和“ - ”，信号参数“ to ”也被保留。该参数可以由超时信号用来重置当前请求的缺省超时值，该值为以毫秒为单位的十进制数值。单个的信号和（或）包定义应该指明包中的一个或多个超时信号是否支持该参数。如果没有指明，版本零中的超时信号被认为不支持该参数，而版本 1（及以上）中的超时信号被认为支持。缺省地，提供的超时值可以被 1000 除并四舍五入到最接近的非零秒数。单个的信号和（或）包也可以定义其它取舍规则。强烈主张所有的新包和超时信号定义都支持“ to ”信号参数。

以下的例子描述了“ to ”参数如何应用一个 6 秒的信号：

```
L/rg(to=6000)

L/rg(to(6000))
```

以下是事件名的例子：

L/hu	挂机转换，在线路包中。

F/0	数字 0，在 MF 包中。
fh	拍叉簧，假定端点的默认包是线路包。
G/rt@0A3F58	在连接“ 0A3F58 ”上的回铃信号。

另外，还可以在请求事件（RequestedEvents）和检测事件（DetectEvents）参数里使用事件的范围和通配符标记来代替单个名字。事件代码“ all ”保留且表示包中的所有事件或信号。“ * ”用来表示全部连接，“ $ ”用来表示当前连接。

下面是这些符号使用的例子：

M/[0-9]	MF 包中的数字 0 到 9。
Hf	拍叉簧，假定线路包是端点的缺省包。
[0-9*#A-D]	DTMF 包（端点的缺省包）中的全部数字和字母。
T/all	中继信息包的所有事件。
R/qa@*	所有连接中的质量告警事件。
R/rt@$	当前连接中的回铃事件。

5.2.5 ConnectionId （连接标识符）

连接标识符编码为十六进制字符串，最大长度 32 个字符。连接标识符作为字符串而不是数值来比较。

5.2.6 ConnectionMode （连接方式）

连接方式描述了连接的操作方式。可能的取值见下表：

方式	意义
M: sendonly	网关只发送分组
M: recvonly	网关只接收分组
M: sendrecv	网关收发分组
M: confrnce	网关将连接设成会议方式
M: inactive	网关不收发分组
M: loopback	网关将电路设成环回方式
M: contest	网关将电路设成测试方式
M: netwloop	网关将连接设成网络环回方式
M: netwtest	网关将连接设成网络一致性检验方式

注意，和连接方式无关，连接上加载的信号仍会导致分组的发送。

连接方式可以通过包来扩展。

5.2.7 ConnectionParameters （连接参数）

连接参数编码为由一对对类型与值组成的字符串。其中，类型或者是两个字母的参数标识符，或者是一个扩展类型；值是一个十进制数。类型和值通过一个等号“ = ”分开。参数之间用逗号分隔。

连接参数值可以包含最多九个数字。如果到达最大值，计数器不再升级，也就是说，它并不绕回（wrap）或溢出。

连接参数类型在下表中指定：

连接参数名	中文名	编码	连接参数值
Packets sent	发送分组数	PS	在此连接上发送的分组数目
Octets sent	发送的八位组数	OS	在此连接上发送的八位组数目
Packets received	接收的分组数	PR	在此连接上接收的分组数目
Octets received	接收的八位组数	OR	在此连接上接收的八位组数目
Packets lost	丢失分组数	PL	在此连接上丢失的分组数，从序列号的失缺中推得
Jitter	抖动	JI	平均的分组间到达抖动，单位毫秒，用整数表示
Latency	延时	LA	平均时延，单位毫秒，用整数表示

连接参数可以采用以下两种不同的方式来扩展：

- 包扩展参数（推荐）
- 供应商扩展参数

包扩展连接参数在包中定义，为推荐的扩展方式，有以下优势：

- 包名有注册机制（IANA）；
- 参数有独立的名空间；
- 扩展能方便地分组；
- 具有通过审查决定是否支持的简单方式；

供应商扩展参数名以“X-”开头，后跟两个或多个字母扩展参数名。

呼叫代理收到不认识的包或供应商连接参数扩展应该忽略这些参数。

连接参数编码的例子如下： P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48

5.2.8 DetectEvents （检测事件）

检测事件（DetectEvents）参数采用由逗号分隔的事件序列来编码，例如：

T: L/hu,L/hd,L/hf,D/[0-9#*]

注意，没有动作可以和事件关联，然而，可以提供事件参数。

5.2.9 EventStates(事件状态)

事件状态（EventStates）参数编码为由逗号分隔的事件序列，比如：ES: L/hu

注意，没有动作可以和其中的事件关联，然而，可以提供事件参数。

5.2.10 LocalConnectionOptions（本地连接选项）

本地连接选项描述可操作参数，这些参数由呼叫代理在连接处理命令中提供给网关，包括：

- 允许的编解码，采用关键字“ a ”，后随冒号及字符串来编码。如果呼叫代理指定一个值序列，这些值应该用分号分隔。对于 RTP，音频编解码应该使用 RTP-AV 简档或它的代替者中定义的编码名，或者 IANA 注册的编码名来指定。作为 MIME 类型注册的非音频媒体必须使用“ <MIME type>/<MIME subtype> ”格式，比如，“ image/t38 ”。
- 打包周期，单位为毫秒，采用关键字“ p ”，后跟冒号及十进制数值来编码。如果呼叫代理指定一个值的范围，该范围表示为由短线（“ - ”）隔开的两个十进制数值（像 SDP 指定“ ptime ”参数一样）。
- 带宽，单位为千比特每秒（kbits/s），采用关键字“ b ”，后跟冒号和十进制数来编码。如果呼叫代理指定一个值的范围，该范围表示为由短线（“ - ”）隔开的两个十进制数值。
- 业务类型参数，编码为关键字“ t ”，后跟冒号以及作为两个十六进制数字编码的值。当连接在 IP 网上传输时，该参数编码为 IP 头中的 8 比特业务类型值参数（也叫 Diffserv 域）。参数中最左边的“ 比特 ”对应于 IP 头中最不重要的比特。
- 回声抑制参数，编码为关键字“ e ”，后跟冒号以及值“ on ”或“ off ”。
- 增益控制参数，编码为关键字“ gc ”，后跟冒号以及可以是关键字“ auto ”或者是表示增益分贝值的十进制数（正值或负值）。
- 静音抑制参数，编码为关键字“ s ”，后跟冒号以及值“ on ”或“ off ”。
- 资源预留参数，编码为关键字“ r ”，后跟冒号以及值“ g ”（保证服务），“ cl ”（控制负荷）或“ be ”（尽最大努力）。
- 密钥参数，编码为关键字“ k ”，后跟冒以及 SDP（RFC2327）的参数“ K ”中定义的密钥说明。
- 网络类型，编码为关键字“ nt ”，后跟冒号和网络类型编码，如“ IN ”（因特网）、“ ATM ”、“ LOCAL ”（用于本地连接）或可能的其他在 IANA 注册过的网络类型。

前三个属性的编码，如果出现的话，要和 SDP 及 RTP 简档相匹配。注意，以上任何属性均是任选的。当出现多个属性时，它们之间用逗号隔开。

本地连接选项的例子如下：

```
L: p:10, a:PCMU

L: p:10, a:G726-32

L: p:10-20, b:64

L: b:32-64, e:off
```

本地连接选项的属性可以通过三种不同的方式来扩展：

包扩展属性（推荐）；

- 供应商扩展属性

- 其它扩展属性

包扩展本地连接选项在包中定义，为推荐的扩展方式，有以下优势：

- 包名有注册机制（IANA）
- 参数有独立的名空间
- 扩展能方便地分组
- 具有通过审查决定是否支持的简单方式

供应商扩展属性由属性名，可能后跟冒号及属性值来构成。属性名以两个字符“ x+ ”开头表示必备扩展，以“ x- ”开头表示可选扩展。当网关收到不认识的必备扩展属性时，它必须拒绝此命令（推荐使用错误代码 525 - unknown extension in LocalConnectionOptions）。

注意，供应商扩展参数使用不可管理的名字空间，会有潜在的名字冲突。因此，主张供应商在他们的扩展参数中包含他们的特定字符串。例如，供应商名称。

最后，考虑到与一些现存实施的后向兼容性，MGCP 也允许其它的扩展属性。但请注意，这些属性扩展并不具有包扩展属性的优势。因而，并不主张这种扩展机制。

5.2.11 MaxMGCPDatagram （最大MGCP数据报）

最大 MGCP 数据报（MaxMGCPDatagram）只能用于审查，也就是说，它是一个有效的请求信息（RequestedInfo）代码且可以作为响应参数来提供。

在响应中，最大 MGCP 数据报（MaxMGCPDatagram）值为十进制数字组成的字符串，最多 9 个字符，不允许以零开头。例如： MD: 8100。

5.2.12 ObservedEvents （观察事件）

观察事件（ObservedEvents）参数提供了一个已经观察到的事件列表。事件代码和通知请求中使用的代码一致。根据数图收集的事件可以组成一个串（但不提倡这样来实现）；如果在数字收集过程中发现了其他事件，它们就作为单独事件的列表来报告。观察事件的例子如下：

```
O: L/hu

O: D/8295555T

O: D/8,D/2,D/9,D/5,D/5,L/hf,D/5,D/5,D/T

O: L/hf, L/hf, L/hu
```

5.2.13 PackageList（包列表）

包列表只能用于审查，也就是说，它是一个有效的请求信息（RequestedInfo）代码且可以作为响应参数来提供。

响应参数包括由逗号分开的所支持的包的列表。列表中的第一个包是缺省包；其中的每个包都由包名，后随冒号以及所支持的包的最高版本号组成。例如：

PL: L:1,G:1,D:0,FOO:2,T:1

本参数为可选参数。

5.2.14 QuarantineHandling（隔离处理）

隔离处理参数由逗号分隔的关键字序列组成：

- 关键字“process”（处理）或“discard”（丢弃）指示对隔离观察事件的处理。默认值为“process”。
- 关键字“step”（一次）或“loop”（多次）指示每个通知请求（NotificationRequest）允许最多通知一次还是多次。默认值为“step”。

举例如下：

Q: loop 处理，多次通知

Q: process 处理，一次通知

Q: discard,loop 丢弃，多次通知

5.2.15 ReasonCode（原因码）

原因代码是由 3 个数字组成的数值，随后可能有空白符和注释。例如：

E: 900 Endpoint malfunctioning

原因代码可以通过包来扩展。

5.2.16 RequestedEvents（请求事件）

请求事件参数提供了被请求的事件序列。每个事件都可能被一个请求动作或动作序列限定。指定动作时，要采用关键字列表来编码，放在圆括号内且动作之间用逗号隔开。各个动作的代码如下：

Action	动作	代码
Notify immediately	立即通知	N
Accumulate	收集	A
Treat according to digit map	根据数图处理	D
Swap	交换	S
Ignore	忽略	I
Keep Signal(s) active	保持信号激活	K
Embedded Notification Request	嵌入通知请求	E

当没有指定动作时，缺省动作是“立即通知”该事件。也就是说，ft 和 ft(N)是一样的。没有列出的事件被忽略（永久事件除外）。

“根据数图处理”动作只能用于指定某些包内数字、字母和数字间定时器，这些包定义数字、字母以及定时器（包括扩展数图字母）的编码。

请求事件列表在一行中编码，多个事件/动作组之间用逗号分开。被请求事件编码举例如下：

R: L/hu(N), L/hf(S,N)

R: L/hu(N), D/[0-9#T](D)

对于“嵌入通知请求”动作，嵌入通知请求参数编码为最多三个由逗号分隔的参数组构成的一个列表。每个参数组以单个字母标识开头，后随一对圆括号括起的参数序列。第一个可选参数组，用字母“R”标识，是嵌入的请求事件参数值；第二个可选参数组，用字母“S”标识，是信号请求参数的嵌入值；第三个可选参数组，用字母“D”标识，是数图的嵌入值（注意，现有的实施和简档或许对这三个参数组的编码顺序可能不同。主张具体实现中接受这样的编码，但不主张产生这样的编码。）

如果没有提供请求事件，参数将设为空值。如果没有提供信号请求，参数设为空值。如果没有数字映射，则使用当前值。下面是嵌入请求的例子：

R: L/hd(E(R(D/[0-9#T](D),L/hu(N)),S(L/dl),D([0-9].[#T])))

R: L/hd(E(R(D/[0-9#T](D),L/hu(N)),S(L/dl)))

一些事件可以由附加的事件参数来限定。这些事件参数由逗号分隔且包含在圆括号中。事件参数也可能包含在双引号中（事实上，由于语法限制，有些事件参数必须包含在双引号中），引号中的字符串采用 UTF-8 编码。

例如，带有事件参数“epar”的事件“foobar”编码为：

R: X/foobar(N)(epar=2)

注意，语法要求对缺省的通知动作也最好包含在请求中。

5.2.17 RequestedInfo（请求信息)

请求信息参数包含由逗号分隔的参数代码序列。例如，要审查通知实体、请求标识符、请求事件、信号请求、数图，隔离处理和检测事件参数，请求信息参数值如下：

F:N,X,R,S,D,Q,T

注意，通常来说，扩展参数也可以被审查。单个的扩展要定义审查操作。

在审查端点命令中，性能请求编码为参数代码“A”，比如： F:A

5.2.18 RequestIdentifier（请求标识符）

请求标识符用于关联通知（NTFY）命令以及触发它的通知请求（RQNT）命令。请求标识符为十六进制字符串，最长 32 个字符。它按照字符串而不是数值来比较。字符“0”保留用作报告永久事件，用于在启动后还没有收到通知请求的情况下。

5.2.19 ResponseAck（响应证实）

响应证实参数用于管理“最多处理一次（at-most-once）”机制。它包含一个由逗号分隔的“证实的事务标识符范围（confirmed transaction-id ranges）”的序列。

每个“证实的事务标识符范围”可以是一个十进制数，精确表明对一个事务的证实；也可以是两个用短线（“-”）分隔的十进制数，描述事务标识符范围中的最低和最高值。

响应证实的例子： K: 6234-6255, 6257, 19030-19044

5.2.20 RestartMethod （重启方法）

重启方法参数编码为关键字“graceful（优雅的）”、“forced（强制的）”、“restart（重启动）”、“disconnected（断开连接）”或“cancel-graceful（取消优雅的）”。

例如：RM: restart。

重启方法可以通过包来扩展。

5.2.21 SignalRequests（信号请求）

信号请求参数提供了被请求的信号名。某些信号，比如录音通知或 ADSI 显示，可以通过附加参数来限定。例如：

- 录音通知的名称和参数
- 要显示的字符串

这些参数放在圆括号内，参数之间用逗号分开。例如：

```
S: L/adsi("123456 Francois Gerard")

S: A/ann(http://ann.example.net/no-such-number.au, 1234567)
```

当使用双引号时，字符串本身为 UTF-8 编码。

当请求多个信号时，信号编码之间用逗号隔开。例如：

```
S: L/adsi("123456 Your friend"), L/rg
```

5.3 MGCP命令与响应

本协议中共定义了九个命令：

- 端点配置命令（EndpointConfiguration）：呼叫代理向网关发送该命令，通知网关“线路侧”所期望的编码特性。
- 通知请求命令（NotificationRequest）：呼叫代理可以向网关发送该命令，通知网关在指定的端点上观察指定的事件，比如摘挂机动作或 DTMF 音。
- 通知命令（Notify）：网关用该命令通知呼叫代理关于其请求事件的发生。
- 建立连接命令（CreateConnection）：呼叫代理使用该命令来建立一个连接，该连接终止于网关内的某个端点。
- 修改连接命令（ModifyConnection）：呼叫代理使用该命令来修改先前建立的连接的相关参数。

- 删除连接命令（DeleteConnection）：呼叫代理使用该命令来删除一个已存在的连接。网关也可以使用该命令来指示不再保持某个连接。
- 审查端点（AuditEndpoint）和审查连接（AuditConnection）命令：呼叫代理可以使用这两个命令来审查一个端点及与端点关联的任何连接的状态。实际上，一般还需要网络管理具备的超出这些命令提供的能力之外的其它能力。这些能力有望通过使用网络管理协议 SNMP 及 MIB 定义来加以支持,这已经超出了本规范所讨论的范畴。
- 正在重启动命令（RestartInProgress）：网关使用该命令通知呼叫代理，它所管理的一组端点正在退出服务或恢复服务。

这些服务允许控制器（通常就是呼叫代理）命令网关建立终止于其上某端点的连接，及接收端点上发生的事件的通知。例如，端点可以是：

- 位于网关内某中继组中的一条指定的中继电路；
- 录音通知服务器所处理的一个指定的录音通知；

下面具体介绍每个命令及其响应，命令名后面圆括号中的列出的参数为命令所携带的参数，命令名前面的参数为命令响应的返回参数，在中括号的参数为可选参数。

5.3.1 EndpointConfiguration

端点配置命令，缩写为 EPCF，用于指示端点接收的信息的编码方式。例如，在某些国际电话配置中，一些呼叫采用 mu 律进行音频编码，而另一些采用 A 律。呼叫代理通过端点配置命令向网关传送这些信息。配置可以根据不同呼叫进行变化，也能用于没有任何连接存在时。

```
ReturnCode,
[PackageList]
<-- EndpointConfiguration(EndpointId,
                          [BearerInformation])
```

端点标识符（EndpointId）是网关中要执行端点配置命令的端点名，不允许使用“ 任意（any of）”通配符。如果使用了“ 所有（all of） ”通配符，本命令适用于所有和该通配符匹配的端点。

承载信息（BearerInformation）定义从线路侧收发的数据的编码信息。这些信息采用子参数序列来编码。本标准中定义的唯一子参数是承载编码，取值为 A 律或 mu 律。子参数可以被扩展。

为了允许扩展且又保持后向兼容，承载信息参数在下述条件下为可选参数：

- 如果不采用扩展参数（包扩展、供应商或其他扩展），承载信息参数是必需的。
- 否则，承载信息参数是可选的。

当被省略时，承载信息必须保持它的当前值。

响应代码（ReturnCode）是网关返回的参数。它指示了命令的执行结果，由一个整形数和可选的注释组成。

包列表（PackageList）是所支持的包序列，也可能包括错误码 518（unsupported package）。

5.3.2NotificationRequest

通知请求命令，缩写为 NQRT，用于请求网关在某端点上产生指定事件时发送通知。例如，请求通知网关内某端点收到的传真信号音。接收该通知的实体也许会决定在该端点的连接上使用不同类型的编码方法，并相应地通过修改连接（MDCX）命令来指示网关。

```
ReturnCode,
[PackageList]
<-- NotificationRequest(EndpointId,
                        [NotifiedEntity,]
                        [RequestedEvents,]
                        RequestIdentifier,
                        [DigitMap,]
                        [SignalRequests,]
                        [QuarantineHandling,]
                        [DetectEvents,]
                        [encapsulated EndpointConfiguration])
```

端点标识符（EndpointId）是网关中要执行通知请求命令的端点名，不允许使用“ 任意（any of）”通配符（即“ $ ”）。

通知实体（NotifiedEntity）为可选参数，指示端点的新“ 通知实体 ”。

请求标识符用于关联该请求与请求触发的通知。该标识符在相应的通知（NTFY）命令中重复出现。

请求事件参数是请求网关检测和报告的一个事件序列，可能由事件参数来限定。例如，这些事件可能包括，传真信号音、一致性检验音或者摘挂机转换。除非另行指明，事件都在端点上检测；然而，有些事件也可以在连接上检测。在请求事件参数中，一个特定的事件不应该出现一次以上。如果该参数被省略，它的缺省值为空。

每个事件和一个或多个动作关联，这些动作是:

- 立刻通知事件，连同收集的观察事件列表（简称为立即通知，Notif）；
- 交换音频（Swap）；
- 在一个事件缓冲中收集事件，但还不通知（简称为收集事件，Accum）；
- 根据数图收集（AccDi）；
- 保持信号激活（KeSiA）；
- 处理嵌入的通知请求（EmbNo）；
- 忽略事件（Ignor）；

要求支持立即通知事件、收集事件、保持信号激活、嵌入的通知请求以及忽略等动作；需要支持在任何能够检测 DTMF 信号的端点上根据数图收集动作 ;除此之外，支持任何其他的动作是可选的。动作可以被扩展。

特定的动作可以缺省地为任何事件设定，尽管有些动作并不对所有事件都有意义。举例来说，

一个带有根据数图收集动作的摘机事件是有效的；但是当摘机事件发生时，当然会立刻触发一个与数图的不匹配。不用说，不主张该类实现。

一些动作能够组合使用，参见下表。其中，“ Y ”意谓两个动作能被组合，“ N ”意谓他们不能够组合：

	Notif	Swap	Accum	AccDi	KeSiA	EmbNo	Ignor
Notif	N	Y	N	N	Y	Y*	N
Swap	-	N	Y	N	N	N	Y
Accum	-	-	N	N	Y	Y	N
AccDi	-	-	-	N	Y	N	N
KeSiA	-	-	-	-	N	Y	Y
EmbNo	-	-	-	-	-	N	N
Ignor	-	-	-	-	-	-	N

注释（*）：如果网关被允许对每个通知请求发送多于一个通知命令，那么，“ 嵌入的通知请求 ”动作只能和“ 立即通知事件 ”动作组合。

如果没有指定动作，缺省为“ 立即通知事件 ”动作。如果指定一或多个动作，则只使用这些动作。当指定两个或多个动作时，每个动作必须根据上表中的定义与其它动作进行组合 --- 每个动作都被假定为同时发生。

如果一个网关收到了一个请求，该请求带有无效的或不支持的动作，或者带有非法动作组合，它必须向呼叫代理返回一个错误（推荐使用错误码 523 - unknown or illegal combination of actions）。

除了命令中的请求事件（RequestedEvents）参数外，一些 MGCP 包可能定义“ 永久事件 ”(通常不建议采用，使用方法见 RFC3435 附录 B)。特定包中的永久事件在实现该包的端点上始终检测。如果一个永久事件不包含在请求事件（RequestedEvents）列表中，且该事件发生了，它总会被像所有的其他事件一样检测和处理，就像永久事件已经被通过“ 立即通知 ”动作请求了一样。即便如此，通知请求（NotificationRequest）命令仍应该代替永久事件来触发通知。因此，非正式的，永久事件可以一直被视为隐含在请求事件（RequestedEvents）列表中，且带有一个“ 立即通知 ”动作，尽管不进行冲突检验。

非永久事件是那些需要明确包含在请求事件（RequestedEvents）参数列表中的事件。请求事件序列完全地替代了以前的请求事件序列。除了永久事件外，只有在请求事件序列中的事件才会被端点检测。如果一个永久事件被包含在请求事件（RequestedEvents）序列中，其中指定的动作将会在请求事件序列存活期内替代相关的永久事件的缺省动作，此后再恢复其缺省事件动作。举例来说，如果“ off-hook ”（摘机）是一永久事件，指定了“ Ignore off-hook ”（忽略摘机）动作，在收到了不带任何摘机事件的新请求时，缺省的“ Notify off-hook ”（通知摘机）动作会被恢复。

网关会合并检测永久事件和请求事件，如果某个事件在两个序列中都不包含，它要被忽略。

呼叫代理可以向网关发送一个带有空请求事件序列（RequestedEvents）的通知请求（NotificationRequest）。例如，呼叫代理在不想收集更多的 DTMF 数子的时候，就可以向网关发送这样的命令。然而，永久事件仍会被检测和通知。

“交换音频”动作可以被用作网关在某个端点上处理多于一个连接的情况。它可应用于呼叫等待，以及其他可能的业务情形。为了避免往返于呼叫代理的消息只是改变端点上哪一个连接接到音频功能上，通知请求（NotificationRequest）可以将一个事件映射到本地音频交换功能，由此，以循环方式来选择“下一个”连接。如果只有一个连接,这个动作等同于空操作。如果有两个以上的连接，它们的顺序并没有规定。如果端点只有两个连接，一个是“inactive”（非激活的）， 另一个是“send/receive”（发/收）方式，那么，“交换音频”会尝试使“send/receive”方式的连接为“inactive”，反之亦然。本规范暂不提供任何有关交换音频的附加细节。

正在检测事件时，如果希望加载信号，可以使用“嵌入的通知请求”动作。该动作可能包括一个新的请求事件（RequestedEvents）、信号请求（SignalRequests）序列和一个新的数图。嵌入的通知请求在语法上就像是收到了一个新的通知请求（NotificationRequest）命令，且具有相同的通知实体（NotifiedEntity）、请求标识符（RequestIdentifier）、隔离处理（QuarantineHandling）和检测事件（DetectEvents）参数。当嵌入的通知请求被激活时，当前拨号字符串（current dial string）被清除；但观察事件序列和隔离缓存不受影响（如果和“立即通知”动作组合，通知动作会清除观察事件列表）。注意，“嵌入的通知请求”动作并不收集触发的事件，但它可以与“收集事件”动作组合使用来达到此目的。如果“嵌入的通知请求”动作失败，将会产生一个嵌入的通知请求失败事件（见附录 B）。

MGCP 的实施应能够至少支持一个层次的嵌入。考虑到这个限制，嵌入的通知请求不必要包含另一个嵌入的通知请求。

号码表是一个可选参数，允许呼叫代理根据哪些数字应该收集来向端点提供一个数字收集方案。如果省略该参数，则使用以前定义的值。如果请求事件（RequestedEvents）参数包含“根据数图收集”动作的请求，数图参数必须显式地或通过以前的命令来定义。这些数字的收集会产生一个数字串。在收到通知请求（RQNT）命令时，该数字串被初始化为空串，因此，后续的通知仅返回在该请求后收到的数字。根据数图收集的数字采用像根据其它任何方式收集的事件一样的方式来报告，且按照它们发生的先后次序排列。因此，其他事件有可能发生在数字序列之间。如果网关被请求“根据数图收集”且网关的相应端点上当前没有数图，网关必须返回一个错误（推荐使用错误码 519-endpoint does not have a digit map）。

下面信号请求（SignalRequests）是一个可选参数，包含网关被要求加载的信号。当省略时，它缺省为空值。当指定多个信号时，信号必须并行加载。除非另行指定，信号适用于端点。然而，一些信号也适用于连接。信号由信号名来标识，信号名是一个事件名，可以通过信号参数来限定。

是信号的例子:

- 振铃（Ringing）
- 忙音（Busy tone）
- 呼叫等待音（Call waiting tone）
- 摘机警告音（Off hook warning tone）
- 连接上的回铃音（Ringback tones on a connection）

信号的名字及描述在相应的包中定义。

缺省情况下，信号适用于端点。如果应用到端点的信号导致了媒体流的产生（音频，视频等），这些媒体流不能被转发端点的任何连接上（缺省规定），与连接的方式无关。举例来说，如果一个呼叫等待音应用到一个正在进行呼叫的端点上，只有正在使用该端点的一方会听到呼叫等待音。然而，个别的信号可能定义不同的行为。

当信号被应用于一个已经收到远端连接描述符（RemoteConnectionDescriptor）的连接时，信号产生的媒体流会在连接上转发，与当前的连接方式无关（包括环回和一致性检验）。如果没有收到远端连接描述符，网关必须返回一个错误（推荐使用错误码 527 - missing RemoteConnectionDescriptor）。注意，该限制不适用于检测连接上发生的事件。

当提供一个信号序列（可能为空）时，该序列完全替代当前激活的超时信号序列。在新列表中没有出现的当前激活超时信号要被停止，且提供的新信号变为激活信号。包含在新列表中的当前激活超时信号要继续保持激活，不被中断；因此，这些超时信号的定时器不会被影响。结果，现在没有其他办法重启当前激活超时信号的定时器，只能先把信号关闭。如果超时信号带有参数，参数的初始值继续有效，不管后来提供了什么值。特定的信号不能在信号请求（SignalRequests）中出现多于一次。注意，如果应用信号 S 到端点、连接 C1 以及连接 C2，将形成三个不同的独立信号。

由信号请求（SignalRequests）参数触发的动作与在事件请求（RequestedEvents）参数中指定的事件收集要保持同步。例如，如果通知请求（NotificationRequest）要求振铃，同时请求事件（RequestedEvents）参数要求检测摘机事件，那么，一旦网关检测到摘机事件，振铃就会停止。正式的定义是,一旦检测到一个请求的事件,所有超时信号的生成都要停止,但带有保持信号激活(Keep signals active）动作的被检出事件是个例外。请求事件（RequestedEvents）和信号请求（SignalRequests）可能使用相同的事件定义；在一种情况下，网关被要求检测某事件的发生；但在另一种情况下，网关被要求产生该事件。端点可以检测或执行的特定的事件和信号由该端点支持的包序列来决定，每个包规定了可以检测或执行的一系列事件和信号。网关的端点被请求检测或执行其不支持的某个包中的事件时，要返回一个错误（推荐使用错误码 518- unsupported or unknown package）。当事件名没有包名来限定时，使用端点的缺省包名。如果事件名没有在缺省包中注册，网关必须返回一个错误（推荐使用错误码 522- no such event or signal）。

呼叫代理可以发送请求信号序列为空的通知请求。例如，当一个超时信号要停止时，就可以这样来实现。

如果信号需要在检测到事件发生时立即应用，可以使用“嵌入的通知请求”动作。嵌入的通知请求可能包括一个新的请求事件（RequestedEvents）、信号请求（SignalRequests）列表和一个新的数图。嵌入通知请求允许呼叫代理提供一个“迷你描述”，该描述随着网关对相关事件的检出立即处理。嵌入的通知请求中指定的任何信号请求会立即开始。必须注意预防呼叫代理和网关间的差异。然而，长期的差异不应该发生，因为新的信号请求完全替代了旧的激活超时事件序列，而简短型信号（BR）总是自行停止。主张限制通断型信号的数量；实施中，呼叫代理最好偶尔打开应该开的通断型信号，关闭应该断的通断型信号。

“忽略事件”动作用于忽略事件，例如，避免一永久事件的通知。然而，缺省地，在事件和激活超时信号之间的同步仍会发生（例如，即使摘机事件的请求带有“忽略”动作，当摘机事件发生时，超时拨号音信号也会停止）。为了防止该同步的发生，可以指定“保持信号激活”动作。

可选的隔离处理（QuarantineHandling）参数指示了对“隔离”事件的处理，也就是说，网关在通知请求命令到达前检测到的事件，这些事件还没有通知到呼叫代理。该参数提供了一组操作选项：

- 是否隔离事件应该被处理或丢弃（缺省为处理）；
- 在对请求响应时，期望网关最多产生一个通知（step by step），或多个通知（loop）。（缺省值是最多一个通知）。

当该参数没有的时候，使用缺省值。

应该注意，隔离处理参数也控制如下事件的处理，该事件在命令收到时已检出并处理，但是并没有通知。

检测事件（DetectEvents）是一个可选参数，可以通过事件参数来限定。它指示网关在隔离期间被请求检测的事件序列。当没有该参数时，隔离期间要检测的事件是上一次收到的检测事件序列中列出的事件。此外，网关也会检测永久事件和那些在请求事件（RequestedEvents）列表中规定的事件，包括那些带有“忽略”动作的事件。

一些事件和信号，例如，线上回铃或质量告警，在端点的连接上而不是端点上被检测或加载。事件名的构成允许呼叫代理指定事件应该被执行或检测的连接。

通知请求（RQNT）命令可能携带一个封装的端点配置（EPCF）命令，它们应用于相同的端点。当这种命令出现时，端点配置命令的参数包含在正常通知请求命令的参数序列中，但端点标识符是个例外，它不能被重复。

被封装的端点配置（EPCF）命令共享通知请求命令（RQNT）执行的结果。如果通知请求命令被拒绝，端点配置命令也不会被执行。

返回码（ReturnCode）是一个由网关返回的参数。它指示了命令的执行结果，且由整数值以及其后可选的注释组成。

包列表（PackageList）是支持的包序列，可能包括错误码 518（unsupported package）。

5.3.3Notify

通知命令，缩写为 NTFY，当触发的事件发生时，观察到的该事件由网关使用通知（NTFY）命令来通知。

```
ReturnCode,
[PackageList]
<-- Notify(EndpointId,
           [NotifiedEntity,]
           RequestIdentifier,
           ObservedEvents)
```

端点标识符（EndpointId）是网关中发出该命令的端点的名称。该标识符必须是完全指定的端点标识符，包括网关的域名。名字本地部份不能使用任何通配符。

通知实体（NotifiedEntity）是一个标识请求的通知实体的参数。该参数等于触发该通知的通知请求命令中的通知实体参数。如果在触发请求命令中没有该参数，相应的通知命令中也不包含该参数。不管通知实体参数的值如何，通知必须被发送给端点的当前通知实体。

请求标识符（RequestIdentifier）是与触发该通知的通知请求命令中的请求标识符参数相同。它用于关联通知和触发它的请求。永久事件被视为包括在上一个通知请求命令中的事件。隐式请求标识符（其值为零“0”）用在刚刚重启动后。

观察事件（ObservedEvents）是网关检测和收集的事件序列。单个通知可能报告一个事件序列，且按照它们检测的先后顺序来报告（FIFO）。

列表中只包含如下事件标识符，这些事件通过通知请求命令的请求事件参数来请求。这包括收集到的（但没通知）或根据数图（仍没有匹配）处理的事件，以及最终触发通知或匹配了数图的事件。应该注意，收到的数字要被加入到观察事件序列中，不管是否是根据数图来收集的。例如，如果一个用户输入数字“ 1234 ”且某事件 E 在数字“ 3 ”和“ 4 ”之间发生，观察事件序列就是“ 1,2,3,E,4 ”。在一个连接上检测到的事件应该包括那个连接名，例如，“ R/qa@0A3F58 ”。

如果观察事件（ObservedEvents）列表达到了端点的能力上限，应该产生一个“ 观察事件满（ObservedEvents Full）”事件（见附录 B，端点应该确保它有能力在观察事件列表中包括这个事件）。如果“ 观察事件满 ”事件不用于触发通知命令，事件处理像以前一样继续（包括数图匹配）；然而，后来的事件将不会被包含观察事件列表中。

响应码（ReturnCode）是呼叫代理返回的一个参数。它指示命令的执行结果，且由一个整形数及其后可能出现的注释组成。

包列表（PackageList）所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.4CreateConnection

建立连接命令，缩写为 CRCX，该命令用于在两个端点间建立一个连接。

```
ReturnCode,
[ConnectionId,]
[SpecificEndPointId,]
[LocalConnectionDescriptor,]
[SecondEndPointId,]
[SecondConnectionId,]
[PackageList]
<-- CreateConnection(CallId,
                     EndpointId,
                     [NotifiedEntity,]
                     [LocalConnectionOptions,]
                     Mode,
                     [{RemoteConnectionDescriptor |
                     SecondEndpointId}, ]
                     [Encapsulated NotificationRequest,]
                     [Encapsulated EndpointConfiguration])
```

连接由它的端点来定义。建立连接命令的输入参数为建立一个网关内的连接提供了必要的数据。

呼叫标识符（CallId）参数指示一个连接所属的呼叫（或会话）的标识符。该参数至少应该，在控制相同网关的所有呼叫代理中是唯一的。属于相同呼叫的所有连接应该共享相同的呼叫标识符。呼叫标识符基本上没其语义上的含义，然而，呼叫标识符可以用于报告和计费过程。它并不影响网关处理连接。

端点标识符（EndpointId）是一个连接端点标识符，用于标识执行建立连接命令的网关中的连接端点。端点标识符可以通过赋值来完全指定（fully-specified），或通过“ 任意 ”（any of）通

配符来部分指定（under-specified）。如果端点为部分指定，端点标识符由网关来赋值，并在响应中由指定端点标识符（SpecificEndpointId）参数返回。当使用了“ 任意 ”（any of）通配符时，分配的端点必须处于服务状态（in service）且其上不能建有任何连接。如果没有这样的端点，要返回错误码 410（no endpoint available）。不能使用“ 所有 ”（all of）通配符。

通知实体（NotifiedEntity）是一个可选参数，为端点指示了一个新的“ 通知实体 ”。

本地连接选项（LocalConnectionOptions）为可选参数，用于呼叫代理指示网关对连接的处理。包含在该参数中的子参数可能是下面的一个或多个（每个子参数不能多于一次出现）：

- 编解码算法：一个或多个编解码，以推荐的顺序排列。考虑到互通，推荐支持 G.711 的 mu 律编码（PCMU）。
- 打包周期：可以指定单个毫秒值或值的范围。打包周期不应该和编解码算法的规范相冲突。如果指定的编解码的帧大小和打包周期不一致，且选用了该编解码算法，则网关有权使用和帧大小一致的打包周期，尽管与指定的不一致。这样做时，网关应该尽可能选择一个和指定的帧大小接近的不为零的打包周期。如果没有指定打包周期，端点应该使用选定的编解码算法的缺省打包周期。
- 带宽：允许的带宽，也就是说，传输层以上的负载净荷加上任何头部开销，例如，IP、UDP 和 RTP。带宽的指定不应该和编解码算法或打包周期的指定相冲突。如果指定了一个编解码算法，那么，网关就有权使用它，即使它导致使用了超出指定值的带宽。任何有关带宽和编解码规定之间的差异都不作为一个错误来报告。
- 业务类型：指示连接使用的业务类别。当没有指定业务类型时，网关使用缺省值零（除非另外提供）。
- 回声抑制：缺省地，电话网关总是在端点上采用回声抑制。然而，对某些呼叫有可能需要取消回声抑制。回声抑制有两个值“ 打开 ”（on）和“ 关闭 ”（off）。如果在没有连接的端点上建立连接时，没有该参数，则端点应该初始化为具有回声抑制；如果在已有连接的端点上建立连接时，没有该参数，则回声抑制保持不变。此后，当检测到带内数据（例如，ITUT 建议 V.8,V.25 及 G.168）时，端点打开或关闭回声抑制。随着带内数据的结束，回声抑制的处理应该恢复到该参数的当前值。建议回声抑制由网关来处理，而不由呼叫代理来指定。
- 静音压缩：电话网关可以进行语音活性检测，避免在静音期间发送数据包。然而，有些情况下（例如，modem 呼叫）需要关闭这种检测。静音压缩参数有两个值，“ 打开 ”（on）和“ 关闭 ”（off）。缺省值为“ 关闭 ”（除非另行提供）。在检测到带内数据时，端点应该关闭静音压缩。随着带内数据的结束，静音压缩的处理应该恢复到该参数的当前值。
- 增益控制：电话网关可以在端点上进行增益控制，以适应信号电平。然而，在某些情况（例如，modem 呼叫）下需要关闭该功能。增益控制参数可以指定为“ automatic ”（自动）或一个增益分贝值。指定的增益将会增加到通过端点发送的媒体上，也会从端点收到的媒体上减去。该参数为可选参数。当端点上没有其他连接且该参数省略时，缺省值为没有增益控制（除非另行规定），这等同于指定零分贝增益值。如果端点上有连接且该参数省略时，增益控制保持不变。检测到带内数据，且需要时，端点应该关闭增益控制。随着带内数据的结束，增益控制的处理应该恢复到该参数的当前值。应该注意，增益控制最好留给网关来处理，因此，不推荐使用该参数。

- RTP 安全：呼叫代理可以请求网关对语音包进行加密。可以通过提供加密规则（见 RFC2327）来做到这一点。缺省地，不进行加密。
- 网络类型：呼叫代理可以命令网关在特定的网络上进行连接。如果省略，该值为相应的网关正在使用的网络类型。
- 资源预留：呼叫代理可以命令网关为连接预留资源。

呼叫代理在命令中指定它所关心的相关子参数，其余的留给网关来设定。对于没有显式指定的上述参数，如果可能的话，网关应该使用缺省值。

方式（mode）指示了连接的这一侧的操作方式。基本的方式是“ send ”、“ receive ” 、“ send/receive ”、“ conference ”、“ inactive ”、“ loopback ”、“ continuity test ”、“ network loopback ”和“ network continuity test ”。注意，端点上加载的信号并不受限于连接方式。某些端点不一定支持所有的方式。如果端点不支持命令指定的方式，会返回一个错误（推荐使用错误码 517 – unsupported mode）。如果一个连接没有收到远端描述符（RemoteConnectionDescriptor）参数，但试图将该连接置为“ send only ”、“ send/receive ”、“ conference ”、“ network loopback ”或者“ network continuity test ”状态，或者要在该连接上加载一个信号（与检测一个事件相反），必须返回一个错误（推荐使用错误码 527 – missing RemoteConnectionDescriptor）。

网关返回一个连接标识符（ConnectionId），由它来唯一标识端点内的连接；它还返回一个本地连接描述符（LocalConnectionDescriptor），该描述符识是一个会话描述，包含了有关连接的信息（在 SDP 中定义），例如，用于媒体流的 IP 地址及端口号。

指定端点标识符（SpecificEndPointId）是一个可选参数，指示了响应端点。它在端点标识符参数使用“ 任意 ”（any of）通配符且命令成功时返回。当指定端点标识符被返回时，呼叫代理应该把它作为到该连接的后续命令的端点标识符。

第二端点标识符（SecondEndpointId）可以用于代替远端连接描述符（RemoteConnectionDescriptor）来建立位于同一个网关内的两个端点间的连接，该连接是一个本地连接。第二端点标识符可以通过赋值来完全指定（fully-specified），或者通过“ 任意 ”（any of）通配符来部分指定（under-specified）。如果第二端点标识符被部分指定，第二个端点标识符由网关来指定，并通过第二端点标识符参数返回。

当第二端点标识符（SecondEndpointId）被指定时，命令建立了两个连接，每一个都可以用修改连接和删除连接命令来单独操纵。除了第一个连接的连接标识符和本地连接描述符外，对建立连接命令的响应还提供了第二连接标识符参数，该参数指示第二个连接。第二个连接为“ send/receive ”方式。

在收到不包括远端连接描述符（RemoteConnectionDescriptor）参数的建立连接（CRCX）请求后，网关处在一种不明确的状态。因为它给出了一个本地连接描述符（LocalConnectionDescriptor）参数，可以接收分组包；但它没有收到另一侧的远端连接描述符参数，不知道收到的分组包是否经过呼叫代理的授权。因此，它必须在两种风险中尝试，也就是说，修剪一些重要的录音通知或聆听荒唐的数据。网关的行为由方式参数值来确定：

- 如果方式为“ ReceiveOnly ”，网关必须接收媒体流，且向端点传送这些媒体流。
- 如果方式为“ Inactive ”、“ Loopback ”或“ Continuity test ”，网关不能向端点传送媒体流。

注意，在这种情况下，方式取值为“ SendReceive ”、“ Conference ”、“ SendOnly ”、“ Network Loopback ”或者“ Network continuity test ”并没有意义，应该被认为是错误，且命令被拒绝（推荐使用错误码 527 – missing RemoteConnectionDescriptor）。

建立连接命令可能包含一个封装的通知请求命令，它应用于端点，此时，必须有请求标识符参数，以及通知请求命令的其它可选参数，但端点标识符参数是个例外，它不能被重复。封装的通知请求命令和连接的建立同时执行。例如，当呼叫代理要开始一个到住宅网关的呼叫时，它可以:

- 要求住宅网关准备一个连接，为确保用户一旦摘机就能开始说话。
- 要求住宅网关开始振铃,
- 要求住宅网关在摘机时通知呼叫代理。

这可以在单个建立连接（CreateConnection）命令中完成，通过在命令中包含带有摘机事件的请求事件参数和带有振铃信号的信号请求参数。

当这些参数提供的时候，建立连接和通知请求必须被同步，意味着两者必须均被接受或拒绝。例如，如果网关没有足够的资源，或者没有从本地网接入得到足够的资源，建立连接命令也许被拒绝；如果用户已经处于在摘机状态，摘机通知请求也可能被拒绝。在该例中，如果连接不能建立，电话就不要振铃；如果用户已经处于摘机状态，连接就不能建立。

如果出现通知实体（NotifiedEntity）参数，则它定义了端点的新的“ 通知实体 ”。

建立连接命令可能携带一个封装的端点配置（EPCF）命令，它应用于端点标识符。当这种命令出现时，端点配置命令的参数包含在正常建立连接命令的参数序列中，但端点标识符是个例外，它不被重复。端点配置命令可能和一个被封装的通知请求命令一起被封装。注意，这两者都仅应用于端点标识符。

被封装的端点配置（EPCF）命令共享通知建立连接命令的结果。如果建立连接命令被拒绝，端点配置命令也不会被执行。

响应码（ReturnCode）是网关返回的一个参数。它指示命令的执行结果，且由一个整形数及其后可能出现的注释组成。

包列表（PackageList）所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.5ModifyConnection（修改连接）

修改连接命令，缩写为 CRCX，该命令用于修改网关连接的特性。这包括本地连接描述符以及远端连接描述符。

```
ReturnCode,
[LocalConnectionDescriptor,]
[PackageList]
<-- ModifyConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     [NotifiedEntity,]
                     [LocalConnectionOptions,]
                     [Mode,]
```

```
[RemoteConnectionDescriptor,]
[Encapsulated NotificationRequest,]
[Encapsulated EndpointConfiguration])
```

除连接标识符（ConnectionId）参数外，该命令使用的参数和建立连接命令的参数基本相同。连接标识符指示端点内的连接，除了本地连接描述符外，该参数也由建立连接命令返回。它唯一地指示了位于端点的关联内的连接。连接建立时所使用的呼叫标识符也必须被包括。

端点标识符必须是一个完全指定的端点表识符。本地名部分不能使用通配符。

修改连接命令可以用于影响一个连接的参数，如下所述：

- 通过远端连接描述符参数来提供关于连接另一端的信息。如果省略该参数，它保持当前值。
- 通过改变方式参数的值，激活或非激活连接。这可以发生在连接期间的任何时刻，且带有任意参数值。如果省略该参数，它保持当前值。
- 通过本地连接选项参数，改变连接的参数，例如，通过转变到不同的编码方案，改变打包周期，或者修改回声抑制的处理等。如果一个或多个本地连接选项参数被省略，那么，网关应该坚持不要改变参数的当前值，除非另外一个参数明确地定义这种改变。例如，编解码变化可能需要静音抑制方面的改变。注意，如果出现远端连接描述符,那么，只有在修改连接命令中提供的本地连接描述符会影响编解码协商。

连接只有在远端连接描述符已经提供的情况下才可能完全激活。然而，只收方式可以在没有该描述符的情况下激活。

如果本地连接参数（例如，RTP 端口）被修改，该命令只返回本地连接描述符。如果只有连接的方式被改变，不需要返回本地连接描述符参数。然而，在命令中包含本地连接选项不是一个本地连接参数发生改变的必备条件。如果一个连接参数（例如，静音抑制）被省略，有可能的话，该参数的以前值将继续保持。如果一个参数的改变需要改变一个或多个未指明参数，那么，网关可以决定为需要改变的未指明参数选择合适的值。例如，这可以发生在打包周期没有指明的情况下；如果新的编解码支持旧打包周期，该参数的值不改变，因为不需要改变；然而，如果不支持旧的打包周期，网关将会选择一个合适的值。

修改连接命令可能包含一个封装的通知请求命令，此时，必须有请求标识符参数，以及通知请求命令的其它可选参数，但端点标识符参数是个例外，它不能被重复。封装的通知请求命令和连接的修改同时执行。例如，当连接接通时，主叫侧网关应该接收到将电路设为收发方式且停止提供回铃音的指令。这可以通过单个修改连接命令完成，方法是在该命令中包含带有挂机事件的请求事件参数和空的信号请求参数（停止提供回铃音）。

当这些参数提供的时候，修改连接和通知请求必须同步，意味着两者必须同时被接受或拒绝。

如果出现通知实体（NotifiedEntity）参数，则它定义了端点的新“通知实体”。

建立连接命令可能携带一个封装的端点配置（EPCF）命令，它应用于相同的端点。当这种命令出现时，端点配置命令的参数包含在正常修改连接命令的参数序列中，但端点标识符是个例外，它不被重复。端点配置命令可能和一个被封装的通知请求命令一起被封装。

被封装的端点配置（EPCF）命令共享修改连接命令的结果。如果修改连接命令被拒绝，端点配置命令也不会被执行。

响应码（ReturnCode）是网关返回的一个参数。它指示命令的执行结果，且由一个整形数及其后可能出现的注释组成。

包列表（PackageList）所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.6DeleteConnection (from the Call Agent)

删除连接命令，缩写为 DLCX，本节是描述由呼叫代理发起的删除连接命令，用于终结一个连接。附带地，它收集连接执行过程中的统计数据。

```
ReturnCode,
ConnectionParameters,
[PackageList]
<-- DeleteConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     [NotifiedEntity,]
                     [Encapsulated NotificationRequest,]
                     [Encapsulated EndpointConfiguration])
```

其中的端点标识符应该是完全指定的，不可使用通配符。

连接标识符指示要删除的连接。还要包括在连接建立时使用的呼叫标识符。

如果出现通知实体（NotifiedEntity）参数，则它定义了端点的新“通知实体”。

在 IP 多播的情况下，连接可以被单个的、独立的删除。然而，在仅有两个端点的单播情况下，删除连接命令必须发给连接涉及到的两个网关。在连接被删除后，先前该连接支持的媒体流不再可用。任何从先前的连接上收到的媒体分组都被简单地丢弃，且不在媒体流上发送新的媒体包。

在连接被删除后，该连接上已经请求的任何环回（loopback）都必须被取消（除非端点上有另一个连接正在请求环回）。

作为对删除连接命令的响应，网关返回一个描述连接统计的连接参数序列。

当连接是一因特网媒体流时，这些参数是：

- 发送的分组数：

自从在连接上开始传送以来，发送方传送的分组总数。在 RTP 的情况下，如果发送方改变了它的同步源标识（SSRC），该计数并不被重新设置。例如，作为修改连接命令的结果。如果连接总是处于“只收”方式且没有信号被加载到该连接上时，计数值是零。

- 发送的八位组数：

自从在连接上开始传送以来，发送方传送的分组包净荷的八位组总数（不包含头和填充字节）。在 RTP 的情况下，如果发送方改变了它的同步源标识（SSRC），该计数并不被重新设置。例如，作为修改连接命令的结果。如果连接总是处于“只收”方式且没有信号被加载到该连接上时，计数值是零。

- 接收的分组数：

自从在连接上开始接收以来，发送方接收的分组包总数。在 RTP 的情况下，如果发送方使用几个不同的同步源（SSRC），该计数包括接收自所有同步源的分组数。如果连接总是处于“只发”方式，计数值是零。

- 接收的八位组数：

自从在连接上开始接收以来，发送方接收的分组包净荷的八位组总数（不包括头和填充字节）。在 RTP 的情况下，如果发送方使用几个不同的同步源（SSRC），该计数包括接收自所有同步源的分组的八位组总数。如果连接总是处于“只发”方式，计数值是零。

- 丢失的分组数：

自从开始接收以来所丢失的的分组包总数。该值被定义为期望收到的分组数减去实际收到的分组数，其中接收的分组数包括任何迟来的或重复的分组。对 RTP 来说，计数值包括来自不同同步源（如果发送者使用了多个 SSRC）的分组。因此，迟来的分组并不统计为丢失，且在有重复的情况下丢失数有可能是负值。期望收到的分组数定义为收到的上一个扩展序号减去收到的初始序号。如果连接总是处于“只发”方式，计数值是零。

- 到达抖动：

媒体包的到达时间的统计变化的估计值，它是以毫秒为单位的无符号整数。对于 RTP 来说，抖动值 J 定义为偏差 D 的平均方差，其中偏差 D 通过接收器与发送器比较一对对分组的差值来获得。详细的算法可在 RFC1889 中找到。如果发送器使用几个不同的同步源（SSRC），该计数包括接收自所有同步源的分组。如果连接总是处于“只发”方式，计数值是零。

- 平均的传输时延：

网络时延的估计值，以毫秒为单位。对 RTP 来说，它是发送方 RTCP 消息指示的 NTP 时戳与接收方的 NTP 时戳之间的偏差的平均值，在消息收到时测量。该平均值通过对所有的估计值求和，再除以收到的 RTCP 消息的数目来计算。当网关的时钟不由 NTP 来同步时，延时值可以作为往返延时的二分之一来计算，通过 RTCP 来测量。当网关无法计算单程或往返时延时，该参数值为零。

这些变量的详细定义，请参见 RFC1889.

当连接通过本地互连建立时，这些参数的意义如下：

- 发送的分组数：

没有意义 – 可以被忽略。

- 发送的八位组数：

本地连接上传送的八位组净荷数。

- 接收的分组数：

没有意义 – 可以被忽略。

- 接收的八位组数：

本地连接上接受的八位组净荷数。

- 丢失的分组数：

没有意义 – 可以被忽略。假定值为零。

- 到达抖动：

没有意义 – 可以被忽略。假定值为零。

- 平均传送时延：

没有意义 – 可以被忽略。假定值为零。

连接参数可以被扩展。由此，参数的意义可以被其它的网络类型进一步进行定义，选择不返回所有的或任何上述参数。

删除连接命令可能包含一个封装的通知请求命令，此时，必须有请求标识符参数，以及通知请求的其它可选参数，但端点标识符参数是个例外，它不能被重复。封装的通知请求命令和连接的删除同时执行。例如，当一个用户挂机事件被通知时，网关应该命令删除连接并且开始检测摘机事件。这可以在单个删除连接命令完成，通过在命令中包含带有摘机事件的请求事件参数和空的信号请求参数。

当这些参数提供的时候，删除连接和通知请求必须被同步，意味着两者必须同时被接受或拒绝。

建立连接命令可能携带一个封装的端点配置（EPCF）命令，它应用于相同的端点。当这种命令出现时，端点配置命令的参数包含在正常删除连接命令的参数序列中，但端点标识符是个例外，它不被重复。端点配置命令可能和一个被封装的通知请求命令一起被封装。

被封装的端点配置（EPCF）命令共享修改连接命令的结果。如果修改连接命令被拒绝，端点配置命令也不会被执行。

响应码（ReturnCode）是网关返回的一个参数。它指示命令的执行结果，且由一个整形数及其后可能出现的注释组成。

包列表（PackageList）所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.7 DeleteConnection (from the gateway)

在很少的情况下，网关需要删除连接，例如，因为丢失了与连接相关的资源，或者因为发现端点不能够或不愿意收发媒体。网关通过使用删除连接命令的一个变体来终结连接：

```
ReturnCode,
[PackageList]
<-- DeleteConnection(CallId,
                     EndpointId,
                     ConnectionId,
                     ReasonCode,
                     Connection-parameters)
```

端点标识符应该是完全指定的，不能使用通配符。

原因码是一个文本字符串，由一个原因代码开始，其后有可能跟描述性字符串。原因码指示了删除连接的原因。

除了呼叫标识符、端点标识符和连接标识符外，网关还发送连接参数，它将作为删除连接的响

应返回给呼叫代理。

响应码是由呼叫代理返回的参数。它指示了命令的执行结果，由一个整数值和其后可选的注释组成。

包列表（PackageList）是所支持的包序列，可以包含错误码 518（unsupported package）。

注意，不主张使用该命令，即使使用，也应作为作为最后的选择。如果连接能够被保持，它的删除就应该由呼叫代理来执行，因为从呼叫代理的地位可以做出更好的判断。

5.3.8 DeleteConnection (multiple connections from the Call Agent)

删除连接命令可以被呼叫代理用于同时删除多个连接。但应注意，这时删除连接命令不能够封装其它命令。该命令用于和一个端点的呼叫相关的所有连接。

```
ReturnCode,
[PackageList]
<-- DeleteConnection(CallId,
                     EndpointId)
```

端点标识符不能使用“ 任意 ”（any of）通配符。端点上存在的具有相应呼叫标识符（CallId）的所有连接都被删除。注意，只要端点标识符有效，即使指定的呼叫标识符上没有连接，该命令仍会成功执行。然而，如果端点标识符无效，命令将会失败。该命令并不返回任何单个的统计值或呼叫参数。

它也能用于一个特定端点上的所有连接：

```
ReturnCode,
[PackageList]
<-- DeleteConnection(EndpointId)
```

端点标识符不能使用“ 任意 ”（any of）通配符。再次说明，即时端点上没有连接，该命令仍会成功执行。

最后，呼叫代理可以利用端点名分级结构的优势，删除属于一组端点的所有连接。在这种情况下，端点标识符的“ 本地名 ”部分将通过“ 所有 ”（all of）通配符来指定，不能使用“ 任意 ”（any of）通配符。例如，如果端点名采用物理接口名和电路号相组合的结构，比如“ X35V3+A4/13 ”，呼叫代理可以用“ 所有 ”（*）通配符来代替电路号，成为“ X35V3+A4/* ”。该“ 通配符 ”命令指示网关删除物理接口“ X35V3+A4 ”上连接的所有电路上的所有连接。

在所有连接被删除以后，连接上被请求的任何环回都要被网关取消。

该命令不返回任何单个的统计或呼叫参数。

响应码是由网关返回的参数。它指示了命令的执行结果，由一个整数值和其后可选的注释组成。

包列表（PackageList）为所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.9 AuditEndpoint

审计端点命令，缩写为 ADEP，呼叫代理用审计端点命令来获得指定端点的状态。

```
ReturnCode,
EndPointIdList,|{
[RequestedEvents,]
[QuarantineHandling,]
[DigitMap,]
[SignalRequests,]
[RequestIdentifier,]
[NotifiedEntity,]
[ConnectionIdentifiers,]
[DetectEvents,]
[ObservedEvents,]
[EventStates,]
[BearerInformation,]
[RestartMethod,]
[RestartDelay,]
[ReasonCode,]
[MaxMGCPDatagram,]
[Capabilities]}
[PackageList]
<-- AuditEndPoint(EndpointId,
                  [RequestedInfo])
```

端点标识符指示需要审计的端点，不能使用“任意”（any of）通配符。“所有”（all of）通配符可用于审计一组端点（不管它们的服务状态）。如果使用该通配符，网关应该通过端点标识符列表（EndPointIdList）参数返回与通配符相匹配的端点标识符序列，它是一个或多个指定端点标识符（SpecificEndpointId），且每个都分别提供。在使用“所有”（all of）通配符的情况下，不应该包括请求信息（RequestInfo）参数（如果包括，必须被忽略）。注意，使用“所有”（all of）通配符可能会产生大量的端点标识符列表（EndPointIdList）。如果认为产生的端点标识符列表（EndPointIdList）太大，网关要返回一个错误（推荐使用错误码 533 - response too large）。

当不指定含有通配符的端点标识符时，请求信息参数（RequestInfo）描述了向指定的端点请求的信息。可以用该命令审查以下端点信息：

RequestedEvents, DigitMap, SignalRequests, RequestIdentifier,

QuarantineHandling, NotifiedEntity, ConnectionIdentifiers,

DetectEvents, ObservedEvents, EventStates, BearerInformation,

RestartMethod, RestartDelay, ReasonCode, PackageList,

MaxMGCPDatagram, 以及 Capabilities.

该列表可以通过扩展参数来扩展。响应将依次包含每个所请求审查的项目的信息。所支持的值为空的参数也要被返回。然而，如果端点被询问一个它所不能够理解的参数，该端点没有必要产生一个错误，但需要在响应时忽略该参数：

- RequestedEvents（请求事件）：端点正在使用的请求事件的当前值，包括每一个事件相关的动作和事件参数，如果没有包括动作，则认为使用缺省动作。永久事件也包含在列表中。如果激活了嵌入的通知请求，请求事件（RequestedEvents）参数反映了在嵌入的通知请求中请求的事件，而不是任何其他的请求事件（不管嵌入与否）。
- DigitMap（数图）：端点正在使用的数图。如果端点没有数图，该参数为空。
- SignalRequests（信号请求）：一个包含当前激活的超时信号、当前打开的通断信号以及任何未处理完的简短信号的列表。已经超时的超时信号和当前正在加载的简短信号不包括在内。包含在原始的信号请求中的任何信号参数都要被包括。
- RequestIdentifier（请求标识符）：该端点收到的上一个通知请求命令的请求标识符（包括封装在其他命令中的通知请求命令）。如果自从重新启动以来没有收到通知请求命令，则返回零值。

- QuarantineHandling（隔离处理）：端点收到的上一个通知请求命令的隔离处理。如果没有包含隔离处理，或者没有收到通知请求，则返回缺省值。
- DetectEvents（检测事件）：最近收到的检测事件的值加上端点上的永久事件。如果没有收到检测事件参数，该列表仅包含永久事件（也可能为空）。
- NotifiedEntity（通知实体）：当前端点的“ 通知实体 ”。
- ConnectionIdentifiers（连接标识符）：特定端点上现存的所有连接的连接标识符序列。
- ObservedEvents（观察事件）：端点上当前观察到的事件列表。
- EventStates（事件状态）：在具有可审查状态的事件中，相应于端点所处的状态的事件，例如，如果端点为摘机状态则为摘机。注意，每个事件的定义要声明该事件本身是否具有可审查的状态。
- BearerInformation（承载信息）：该端点上一次收到的承载信息参数值（包括承载信息被预置的情况）。如果端点没有收到承载信息参数且没有预置值时，该参数为空。
- RestartMethod（重启动方法）：端点处于服务状态且操作正常时发生的“ 重起动 ”，或者端点正在进入服务状态的过程中（一个非零的重启动延时指示后者）的方法。否则，端点发送上一个正在重启动命令中的重启动方法参数的值。注意，一个“ 断开的 ”端点在断开期间只报告“ 断开 ”，一旦不再断开，则报告“ 重起动 ”。像似地，不报告“ 取消优雅地 ”，但可能报告“ 优雅地 ”。
- RestartDelay（重启动时延）：在响应时，如果正在重启动命令要被端点发送，规定其中的重启动参数的值。如果命令不包含该参数，则其值为零。
- ReasonCode（原因码）：网关的端点发送的上一个正在重启动命令或删除连接命令中的原因码参数值，如果端点的状态正常，则用特殊值 000。
- PackageList（包列表）：端点支持的包，包括包的版本号。为可选参数。
- MaxMGCPDatagram（最大 MGCP 数据报）：端点可以接收的最大尺寸的 MGCP 数据报，以字节为单位。该值不包括任何低层的开销。该参数为可选参数。如果没有返回该值，则假定为缺省值。

- Capabilities（性能）：端点的性能和本地连接选项类似，但包括包和连接方式。它也可以被扩展。如果报告了任何未知的性能，它们必须被忽略。如果有必要指定一些参数（例如，静音抑制）只和某些编解码算法兼容，那么，网关必须返回几个性能序列，每个包含：

- 压缩算法：支持的编解码序列。性能序列中的其他参数将应用该序列中指定的所有编解码。
- 打包周期：可以指定一个值或一个范围。
- 带宽：可以指定相应于打包周期的一个值或一个范围（假定没有静音抑制）。
- 回声抑制：端点上是否支持回声抑制。
- 静音抑制：是否支持静音抑制。
- 增益控制：是否支持增益控制。
- 业务类型：是否支持业务类型。
- 资源预留：是否支持资源预留。
- 安全性：是否支持媒体安全。
- 网络类型：支持的网络类型。
- 包：支持包的列表，其中的第一个包为缺省包。
- 方式：支持的连接方式的列表。

呼叫代理可以通过使用审计连接命令来获得有关连接的进一步信息。

如果没有请求信息且端点标识符为一个有效的端点，网关返回一个肯定响应。

响应码是由网关返回的参数。它指示了命令的执行结果，由一个整数值和其后可选的注释组成。

注意，包列表（PackageList）也可以包含错误码 518（unsupported package）。

5.3.10 AuditConnection

审计连接命令，缩写为 ADCX，呼叫代理用审计连接命令来获得连接上的参数。

```
ReturnCode,
[CallId,]
[NotifiedEntity,]
[LocalConnectionOptions,]
[Mode,]
[RemoteConnectionDescriptor,]
[LocalConnectionDescriptor,]
[ConnectionParameters,]
[PackageList]
<-- AuditConnection(EndpointId,
                    ConnectionId,
                    RequestedInfo)
```

端点标识符参数指示所处理连接的端点，不能使用通配符。

连接标识符参数是位于指定端点的关联内且要审查的连接的标识符。

请求信息（可能为空）描述了位于指定的端点标识符内的连接标识符的请求信息。该命令可以请求以下连接信息：

```
CallId, NotifiedEntity, LocalConnectionOptions, Mode,
RemoteConnectionDescriptor, LocalConnectionDescriptor,
ConnectionParameters
```

审计连接的响应将依次包括每一个正在请求审查条目的信息：

- 呼叫标识符（CallId），连接所属呼叫的呼叫标识符。
- 通知实体（NotifiedEntity），连接的当前“通知实体”。注意，它和端点的“通知实体”相同。
- 本地连接选项（LocalConnectionOptions），实际提供给连接的最近的本地连接选项参数（忽略来自命令但不改变该值的本地连接选项）。注意，不包括来自最近的本地连接选项省略的缺省参数。通过修改连接命令没有被修改的本地连接选项， 如果它们在以前的修改命令中曾经包含，不管它们在最近的本地连接选项参数中是否提供，这些连接选项仍要包含。
- 方式（Mode），当前的连接方式。
- 远端连接描述符（RemoteConnectionDescriptor），提供给网关的连接的远端连接描述符。
- 本地连接描述符（LocalConnectionDescriptor），提供给网关的连接的本地连接描述符。
- 连接参数（ConnectionParameters），连接上的连接参数的当前值。

如果没有请求信息且端点标识符有效，网关只要检查连接的存在，如果存在，则返回肯定证实。注意，根据定义，端点处于服务状态才有可能导致这种情况的发生，因为退出服务的端点没有任何连接。

响应码是由网关返回的参数。它指示了命令的执行结果，由一个整数值和其后可选的注释组成。

包列表（PackageList）是所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.11 RestartInProgress

正在重启动命令，缩写为 RSIP，用于由网关发出一个端点或一组端点被投入服务或退出服务的信息。

```
ReturnCode,
[NotifiedEntity,]
[PackageList]
<-- RestartInProgress(EndPointId,
                      RestartMethod,
                      [RestartDelay,]
                      [ReasonCode])
```

端点标识符指示被投入服务或退出服务的端点。“所有”通配符可以用于应用该命令到相同的呼叫代理管理的一组端点上，例如，指定接口上的所有端点，甚至特定网关上的所有端点。“任一”通配符不能使用。

重启动方法参数指定了重启动类型。定义了以下值：

- 一个“ 优雅的 ”重启动方法指示端点将在指定的时延后被退出服务。已经建立的呼叫暂时不受影响，但呼叫代理应该限制建立新的连接，且尽量优雅地释放存在的连接。
- 一个“ 强制的 ”重启动方法指示指定的端点被立即退出服务。已经建立的连接被释放。
- 一个“ 重启动 ”方法指示端点上的业务在指定的“ 重启动时延 ”后将被恢复，也就是说，端点将投入服务。端点将处于缺省状态且其上没有建立连接。
- 一个“ 断开的 ”方法指示端点变为断开状态且现在正试图建立连接。“ 重启动时延 ”指定已经被断开的端点的秒数。已经建立的连接不受影响。
- 一个“ 取消优雅的 ”方法指示网关正在取消先前发送的“ 优雅的 ”重启动方法。该端点仍在服务中。

重启动方法的列表可以被扩展。

可选的“ 重启动延时 ”参数用秒数来表示。如果没有该秒数，延时值必须被认为是零。在“ 优雅的 ”方法下，零延时指示呼叫代理要等待所有存在的连接自然释放，且不建立新连接。在“ 强制的 ”和“ 取消优雅的 ”方法下，重启动延时一直被认为是零，因此，“ 重启动延时 ”参数不要和这两个重启动方法组合使用。当网关发送带有非零重启动延时的“ 重启动 ”或“ 优雅的 ”方法的正在重启动命令时，网关应该在“ 重启动时延 ”过去后发送更新的正在重启动消息。

带有值为零的重启动时延的“ 重启动 ”方法指示服务已经恢复。它典型地发生在网关重启动以后。为了减轻作为重启动的结果而导致的网关 IP 地址变化的影响，呼叫代理可能需要刷新它的网关域名 DNS 缓冲存储器或者通过 DNS 来解析网关域名，而不管当前对重启动网关记录的 DNS 资源的存活周期。

可选的原因码参数指示了重启动的原因。

作为对呼叫代理的礼貌，当网关退出服务时，例如，被停止或者被网管系统退出服务，它们应该发送一个“ 优雅的 ”或“ 强制的 ”正在重启动消息（对相关的端点），然而呼叫代理不能总依赖接收这样的消息。当网关投入服务时，根据重启动过程，它们必须给呼叫代理发送一个带有“ 重启动 ”方法的正在重启动命令，且重启动零延设为零，呼叫代理可以依赖接收该消息。根据“ 断开 ”过程，网关必须给它们的当前“ 通知实体 ”发送一个“ 断开 ”方法的正在重启动命令。

正在重启动命令将被发送到当前正在处理的端点的“ 通知实体 ”。期望在重启动后有一个缺省的呼叫代理（即“ 通知实体 ”），该缺省的呼叫代理一直是端点的“ 通知实体 ”。当一个网关内的多个端点重启动且这些端点由相同的呼叫代理来管理时，网关应该充分利用通配符的好处来减少产生的重启动消息的数量。

原因码是呼叫代理返回的一个参数。它指示了命令的执行结果，且由一个整数和其后可能的注释组成。

作为对来自呼叫代理的正在重启动命令的响应，通知实体可能被返回。这通常仅在对“ 重启动 ”或“ 断开的 ”方法进行响应时才发生：

- 如果响应值指示成功（响应码 200 – transaction executed），重启动成功完成，且返回的通知实体是该端点的新“ 通知实体 ”。
- 如果来自呼叫代理的响应指示一个错误，重启动没有完全成功。如果通知实体参数包含在返

回的响应中，它指明了该端点的一个新“ 通知实体 ”，它必须用于重新尝试重启动过程。这只能通过错误码 521（endpoint redirected）来做。

注意，上述在响应中返回通知实体参数的行为只对正在重启动命令的响应来定义，不应该作为对其他命令的响应来使用。任何其他的行为都未定义。

包列表（PackageList）是所支持的包序列，可以包含错误码 518（unsupported package）。

5.3.12 CreateConnection Response

对建立连接消息的响应行由成功的响应码 200，后跟连接标识符参数构成。本地连接描述符通过肯定响应来传送。本地连接描述符被编码为“ 会话描述 ”（SDP，RFC2327）。它被一个空行与响应头分开，例如：

```
200 1204 OK
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

当先前发送了临时响应时，最终响应可以进一步包含响应证实参数，例如：

```
200 1204 OK
K:
I: FDE234C8

v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

最终响应应该再通过响应证实来证实，例如：

```
000 1204
```

5.3.13 ModifyConnection Response

在成功的修改连接消息的情况下，如果该修改命令导致了会话参数的修改（例如，只改变连接

方式并不会改变会话参数），响应行后要跟本地连接描述符。本地连接描述符被编码为“ 会话描述 ”（SDP，RFC2327）。它通过一个空行与响应头分开，例如：

```
200 1207 OK
v=0
o=- 25678 753849 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

当先前发送了临时响应时，最终响应可以进一步包含响应证实参数，例如：

```
200 1207 OK
K:
```

最终响应应该再通过响应证实来证实，例如：

```
000 1207 OK
```

5.3.14 DeleteConnection Response

根据删除连接消息的变体，响应行可以后跟一个连接参数参数行，例如：

```
250 1210 OK
P: PS=1245, OS=62345, PR=780, OR=45123, PL=10, JI=27, LA=48
```

5.3.15 NotificationRequest Response

成功的通知请求响应不包括任何其他的响应参数。

5.3.16 Notify Response

成功的通知响应不包括任何其他的响应参数。

5.3.17 AuditEndpoint Response

在成功地审查端点的情况下，响应行可能后跟每一个被请求参数的信息，每个参数出现在一个独立的行中。当前没有赋值的参数（例如，数图）也要提供，但值为空。每个通过通配符“ 扩展 ”的本地端点名将出现在单独的行中，使用“ 指定端点标识符（SpecificEndPointId） ”参数进行编码，例如：

```
200 1200 OK
Z: aaln/1@rgw.whatever.net
Z: aaln/2@rgw.whatever.net
```

当连接标识符被审查且端点上有多个连接时，应该返回一个以逗号分隔的连接标识符序列，例

如：

```
200 1200 OK
I: FDE234C8, DFE233D1
```

作为选择，也可以返回多个连接标识符参数行。这两种形式不应该混合使用，尽管那样做并不构成错误。

当审查能力时，可以响应多个能力参数行，例如：

```
200 1200 OK
A: a:PCMU;G728, p:10-100, e:on, s:off, t:1, v:L,
    m:sendonly;recvonly;sendrecv;inactive
A: a:G729, p:30-90, e:on, s:on, t:1, v:L,
    m:sendonly;recvonly;sendrecv;inactive;confrnce
```

注意：上述例子中使用回车仅由于格式原因，实际命令的编码中是不允许的。

5.3.18 AuditConnection Response

在成功地审查连接的情况下，响应可能后跟每一个被请求参数的信息。当前没有赋值的参数（例如，数图）也要提供。连接描述符总是放在最后且每个之前都有一个空行，例如：

```
200 1203 OK
C: A3C47F21456789F0
N: [128.96.41.12]
L: p:10, a:PCMU;G728
M: sendrecv
P: PS=622, OS=31172, PR=390, OR=22561, PL=5, JI=29, LA=50

v=0
o=- 4723891 7428910 IN IP4 128.96.63.25
s=-
c=IN IP4 128.96.63.25
t=0 0
m=audio 1296 RTP/AVP 96
a=rtpmap:96 G726-32/8000
```

如果本地和远端连接描述符都被提供，本地连接描述符将会出现在前面。如果请求了一个连接描述符，但它在被审查的连接上并不存在，该连接描述符将只呈现为 SDP 协议的版本行（“ v=0 ”），例如：

```
200 1203 OK
v=0
```

5.3.19 RestartInProgress Response

一个成功的正在重启动响应可以包含通知实体参数，但不包含任何其他响应参数。

一个对正在重启动命令的响应码为 521 的响应必须包括一个通知实体参数。该参数包含在第一个呼叫代理将端点重定向到另一个呼叫代理时，另一个要联系的呼叫代理的名字，例如：

```
521 1204 Redirect
N: CA-1@whatever.net
```

6 响应码、错误码、原因码

6.1 响应码和错误码

所有的 MGCP 命令都要被证实。证实包含一个响应码，它指示了命令的执行状态。响应码是一个整数，且按照范围进行分类定义：

- 000 到 099 表示响应证实
- 100 到 199 表示临时响应
- 200 到 299 表示成功完成
- 400 到 499 表示临时错误
- 500 到 599 表示永久错误
- 800 到 899 是包特定的响应代码
- 关于临时错误（4XX 错误码）与永久错误（5XX 错误码）的对比，概括说明如下：
- 如果呼叫代理收到一个临时错误，就可以预期将来端点还有可能接受类似的请求。在某些情况下，这有可能需要端点所处的环境状态发生改变（例如，在错误码 401 或 402 的情况下的叉簧状态；在错误码 403 的情况下的资源可用性；在错误码 404 的情况下的带宽可用性）。
- 永久错误（错误码 500 到 599）指示一个或多个永久条件，由于协议错或者端点和呼叫代理间的不兼容，或者因为某些呼叫代理不可控的错误条件。例如，协议错误，请求端点不存在的能力，请求中缺少信息或含有不正确的信息，或者任何其他不随时间消失的条件。

当前已经定义的值如下：

000 Response Acknowledgement.

响应证实。

100 The transaction is currently being executed. An actual completion message will follow later.

事务当前正在处理。实际完成消息在后续消息中。

101 The transaction has been queued for execution. An actual completion message will follow later.

事务已经排队等候执行。实际完成消息在后续消息中。

200 The requested transaction was executed normally.

请求的事务正常执行。该响应码可以用于对任何命令的成功响应。

250 The connection was deleted.

连接被删除。该响应码仅用于对删除连接命令的成功响应。

400 The transaction could not be executed, due to some unspecified transient error.

由于临时错误，事务无法处理。

401 The phone is already off hook.

电话已经摘机

402 The phone is already on hook.

电话已经挂机

403 The transaction could not be executed, because the endpoint does not have sufficient resources at this time.

事务无法处理，因为此时端点没有足够的资源

404 Insufficient bandwidth at this time.

此时带宽不足

405 The transaction could not be executed, because the endpoint is "restarting".

事务无法处理，因为端点正在重启动

406 Transaction time-out.

事务超时。事务没有在合理的时间内完成，已经被退出

407 Transaction aborted.

事务退出。事务被某些外部动作退出，例如，修改连接命令被删除连接命令退出

409 The transaction could not be executed because of internal overload.

事务无法处理，因为内部过载

410 No endpoint available.

没有可用端点。使用了一个有效的“任一”（$）通配符，但是没有可用端点来满足该请求

500 The transaction could not be executed, because the endpoint is unknown.

事务无法处理，因为含有未知端点。

501 The transaction could not be executed, because the endpoint is not ready.

因端点未准备好，事务无法处理。包括端点退出服务的情况

502 The transaction could not be executed, because the endpoint does not have sufficient resources (permanent condition).

因端点资源不足（永久条件），事务无法处理

503 "All of" wildcard too complicated.

“ 所有 ”通配符太复杂

504 Unknown or unsupported command.

未知的或不支持的命令

505 Unsupported RemoteConnectionDescriptor.

不支持的远端连接描述符。应该用于不支持远端连接描述符中的一个或多个强制性参数或值

506 Unable to satisfy both LocalConnectionOptions anRemoteConnectionDescriptor.

不能满足本地连接选项和远端连接描述符。应该用于本地连接选项和远端连接描述符包含若干个互相冲突或不能同时支持的必备参数或值的情况（除了编解码协商失败—见错误码534））

507 Unsupported functionality.

不支持的功能。不支持命令执行需要的某些未指明功能。注意，已经定义了几个用于指明的不支持功能域的错误码（例如，508，511 等），因此，该错误码应该仅用于没有其他更明确的不支持功能错误码的情况）

508 Unknown or unsupported quarantine handling.

未知的或不支持的隔离处理

509 Error in RemoteConnectionDescriptor.

远端连接描述符错误。用于远端连接描述符中存在语法或语义错误时

510 The transaction could not be executed, because some unspecified protocol error was detected.

因发现不明确的协议错误，事务无法处理。该错误的自动恢复将会非常困难，因此，该错误码仅作为最后的手段

511 The transaction could not be executed, because the command contained an unrecognized extension.

因命令包含不认识的扩展，事务无法处理。该错误码应该用于不支持的必备参数扩展（X+）

512 The transaction could not be executed, because the gateway is not equipped to detect one of the requested events.

因网关没有检测所请求事件的配置，事务无法处理

513 The transaction could not be executed, because the gateway is not equipped to generate one of the requested signals.

因网关没有生成被请求信号的配置，事务无法处理

514 The transaction could not be executed, because the gateway cannot send the specified announcement.

因网关不能发送指定的录音通知，事务无法处理

515 The transaction refers to an incorrect connection-id (may have been already deleted).

事务涉及到一个错误的连接标识符（可能已被删除）

516 The transaction refers to an unknown call-id, or the call-id supplied is incorrect

(e.g., connection-id not associated with this call-id).

事务涉及一个未知的呼叫标识符，或呼叫标识符不正确（例如，连接标识符没有与该呼叫标识符关联）

517 Unsupported or invalid mode.

不支持的或无效的方式

518 Unsupported or unknown package.

不支持或未知的包。建议在响应中包含所支持包的包列表参数，尤其该响应由呼叫代理

产生时。

519 Endpoint does not have a digit map.

端点没有号码表。

520 The transaction could not be executed, because the endpoint is "restarting".

因端点正重启，事务无法处理。在大多情况下，这是一个临时错误，应该使用错误码 405。保留该错误码是为了后向兼容。

521 Endpoint redirected to another Call Agent.

端点重定向到另一个呼叫代理。当发送对正在重启动命令的响应时，相关的重定向行为才是明确定义的

522 No such event or signal.

没有该事件或信号。请求涉及到的事件或信号在相关的包中没有定义（也可以是缺省包）

523 Unknown action or illegal combination of actions.

未知或非法的动作组合

524 Internal inconsistency in LocalConnectionOptions.

本地连接选项内部矛盾

525 Unknown extension in LocalConnectionOptions.

本地连接选项中未知的扩展。该错误码应该用于不支持的必备供应商扩展（X+）

526 Insufficient bandwidth.

带宽不足。在有临时错误的情况下，应该使用错误码 404。

527 Missing RemoteConnectionDescriptor.

缺少远端连接描述

528 Incompatible protocol version.

不兼容的协议版本

529 Internal hardware failure.

内部硬件错误

530 CAS signaling protocol error.

CAS 信令协议错误

531 Failure of a grouping of trunks (e.g., facility failure).

中继分组失败（比如设备失败）

532 Unsupported value(s) in LocalConnectionOptions.

本地连接选项中不支持的值

533 Response too large.

响应太大

534 Codec negotiation failure.

变解码协商失败

535 Packetization period not supported.

不支持的打包周期

536 Unknown or unsupported RestartMethod.

未知的或不支持的重启动方法

537 Unknown or unsupported digit map extension.

未知的或不支持的数图扩展

538 Event/signal parameter error (e.g., missing, erroneous, unsupported, unknown, etc.).

事件/信号参数错误（例如，缺少，错误，不支持，未知等）

539 Invalid or unsupported command parameter.

无效的或不支持的命令参数。该错误码应该仅用于参数既不是一个包也不是供应商扩展参数时

540 Per endpoint connection limit exceeded.

超过了每个端点的连接限制

541 Invalid or unsupported LocalConnectionOptions.

无效的或不支持的本地连接选项。该错误码应该仅用于本地连接选项既不是一个包也不是供应商扩展的本地连接选项时）

响应码可以在协议的将来版本中扩展。具体执行中如果收到了未知的或不支持的响应码，应该对响应码进行以下处理：

未知的 0xx 作为 000 来处理；

未知的 1xx 作为 100 来处理；

未知的 2xx 作为 200 来处理；

未知的 3xx 作为 521 来处理；

未知的 4xx 作为 400 来处理；

未知的 5xx-9xx 作为 510 来处理；

6.2 原因码

当删除连接时，网关使用原因码通知呼叫代理删除连接的原因。正在重启动命令也可能用原因码来通知呼叫代理重启动原因。

原因码是一个整数，现在已经定义了如下值：

000 Endpoint state is normal (this code is only used in response to audit requests).

端点状态正常（只用于审查请求命令的响应）

900 Endpoint malfunctioning.

端点故障

901 Endpoint taken out-of-service.

端点被退出服务

902 Loss of lower layer connectivity (e.g., downstream sync).

丢失低层连接（比如下行流同步）

903 QoS resource reservation was lost.

QoS 资源预留丢失

904 Manual intervention.

人工干预

905 Facility failure (e.g., DS-0 failure).

设施错误（例如，DS-0 错误）

原因码可以被扩展。

7 连接描述

7.1 使用本地连接选项和连接描述符

正常的建立一个双向连接的顺序包括以下 3 个步骤：

1) 呼叫代理要求一个网关在一个端点上“ 建立一个连接 ”。网关为连接分配资源，并在对命令的响应中提供“ 会话描述 ”信息（作为它的本地连接描述符）。在会话描述中包含了第三方所必需的向新建连接发送数据包的信息。

2) 然后，呼叫代理要求另一个网关在一个端点上“ 建立一个连接 ”。命令里包含第一个网关提供的“ 会话描述 ”信息（现在作为远端连接描述符）。网关为该连接分配资源，并在响应消息里提供自己的“ 会话描述 ”信息（本地连接选项）。

3) 呼叫代理向第一个端点发送“ 修改连接 ”命令，命令中包含第二个“ 会话描述 ”信息（现在作为远端连接描述符）。修改连接完成后，在两个端点之间就可以进行双向通信了。

因此，当呼叫代理发送建立或修改连接命令时，有三个参数决定连接所支持的媒体：

- 本地连接选项（LocalConnectionOptions）：由呼叫代理提供，用于控制网关上连接所使用的媒体参数。当提供时，网关必须遵从这些媒体参数，直到连接被删除或者收到带有新媒体参数（本地连接选项或远端连接描述符）的修改连接命令。
- 远端连接描述符（RemoteConnectionDescriptor）：由呼叫代理提供，用于传达连接的另一侧所支持的媒体参数。当提供时，网关必须遵从这些媒体参数，直到连接被删除或者收到带有新媒体参数（本地连接选项或远端连接描述符）的修改连接命令。
- 本地连接描述符（LocalConnectionDescriptor）：由网关提供给呼叫代理，用于传达网关为该连接所支持的媒体参数。当提供时，网关必须遵从这些媒体参数，直到连接被删除或者网关发送了该连接的一个新的本地连接描述符。
- 在决定需要提供本地连接描述符中的哪个编解码算法时，网关必须考虑三个编解码列表：
- 当前命令中的本地连接选项允许的编解码列表（或者显式地通过编码方法，或者隐式地通过带宽和/或打包周期来指示）。
- 当前命令中远端连接描述符内的编解码列表。
- 网关能够为该连接支持的一个内部编解码列表。对特定的连接，网关可以支持一个或多个编解码

编解码选择（包括所有相关的媒体参数）可以描述为以下步骤：

1) 通过内部编解码列表和本地连接选项允许的编解码列表的交集形成一个核准的编解码列表。如果在当前命令中没有提供本地连接选项，则认可的编解码列表为内部编解码列表。

2) 如果核准的编解码列表为空，则发生编解码协商失败，产生一个错误响应（推荐使用错误码 534 - codec negotiation failure）。

3) 否则，通过核准的编解码列表和远端连接描述符允许的编解码列表的交集形成一个协商的编解码列表。如果在当前的命令中没有提供远端连接描述符，协商的编解码列表只能是核准的编解码列表。

4) 如果协商的编解码列表为空，则发生编解码协商失败，产生一个错误响应（推荐使用错误码 534 - codec negotiation failure）。

5) 否则，便解码协商成功，通过本地连接描述符返回协商的编解码列表。

注意，本地连接选项和远端连接描述符均包含按优先递减顺序排列的编解码列表。当两者都在当前命令中出现时，网关必须遵从本地连接选项中提供的优先顺序。

7.2 资源预留

网关可以被要求在特定的连接上进行资源预留，例如，通过 RSVP。当需要预留时，呼叫代理将指定使用的资源预留简档（profile），为“ 受控负载 ”或者“ 保证服务 ”。不进行资源预留可以通过“ 尽力而为 ”服务来指明，它也是在建立连接命令中该参数的缺省值。对于修改连接命令，缺省情况为仅保持当前值。当连接上请求资源预留时，网关将：

- 在连接处于“ sendonly ”、“ sendrecv ”、“ conference ”、“ network loopback ”或“ network continuity test ”方式时（如果已经收到合适的远端连接描述符），开始发出 RSVP 的“ PATH ”消息。
- 如果连接处于“ receiveonly ”、“ sendrecv ”、“ conference ”、“ network loopback ”或“ network continuity test ”方式时，一旦收到“ PATH ”消息，开始发出 RSVP 的“ RESV ”消息。

RSVP 过滤器将从连接的特性中导出。RSVP 响应简档（profiles）将从连接的编解码、带宽和打包周期中导出。

7.3 SDP的应用

呼叫代理通过 MGCP 来向端点提供连接描述参数，比如 IP 地址，UDP 端口和 RTP 简档。这些描述将遵循会话描述协议 SDP（RFC2327）。MGCP 的实施要求完全能够解析任何符合 SDP 协议的消息，必须发送严格符合 SDP 标准的会话描述。

尤其应该注意：

- 发起 Origin（“ o= ”），
- 会话名称（“ s= ”），和
- 时间 Time active（“ t= ”）

在 RFC2327 中都是必备的。虽然它们对 MGCP 来说几乎没有用，但仍要提供。在提供时，建议使用以下值：

Origin
o = <username> <session-id> <version> <network-type> <address-type> <address>

- username 应该被设置为短线（“ - ”）。

- 建议 session-id 使用 NTP 时戳。
- 版本号 version 必须随着 SDP 的每一次改变而增加。建议初始化为 NTP 时戳。
- network-type 定义了网络类型。对于 RTP 会话，网络类型应该时“ IN ”。
- address-type 定义了地址类型。对于 RTP 会话，地址类型应该是“ IP4 ”（或“ IP6 ”）。
- address 应该时和连接信息（“ c= ”）域提供的地址信息相同。

Session Name
s = <session name>
会话名称应该设置为短线（“ - ”）。

Time active
t = <start-time> <stop-time>

* start-time 可以设为零。
* stop-time 应该设为零。

以上三个参数在收到时都可以被忽略。

为了进一步适应 MGCP 可扩展性原则，鼓励具体实施中支持 PINT“ a=require ”属性（请参考 RFC2848）。

SDP 的使用依赖于正在建立的会话类型。下面，本协议描述 SDP 用于音频业务（使用 RTP/AVP）或用于本地互连的具体用法。如果描述内容和 SDP 冲突，以 SDP 规范为准。

7.3.1 SDP用于音频业务

在电话网关中，仅需要描述仅使用一种媒体（音频）的会话。SDP 的这种应用在 RFC2327 中明确定义。以下为该类应用的例子：

```
v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0 96
a=rtpmap:96 G726-32/8000
```

7.3.2 SDP用于本地连接

当 MGCP 用于在网关内部建立连接时，SDP 格式用于对该连接的参数进行编码。连接和媒体参数使用如下：

- 连接参数（c=）将指示连接是本地的，其中使用关键字“ LOCAL ”来作为网络类型，关键字

“ EPN ”作为地址类型，端点的本地名作为连接地址。

- “ m=audio ”参数指示端口号，它一直被设为零；协议的类型，一直被设为关键字 LOCAL；编码类型，使用和 RTP/AVP 简档（RTP 静荷数）相同的约定。编码类型应该被设为零（PCMU）。

会话级别的属性指示了可能提供的连接。这使得端点支持多个 LOCAL 连接。使用该属性是可选的，对于仅支持单个 LOCAL 连接的端点是不需要的。该属性定义如下：

a=MGCPlocalcx:<ConnectionID>

MGCP 本地连接属性是一个会话级且与大小写无关的属性，它用于指示 SDP 应用的 MGCP 本地（LOCAL）连接，该连接位于连接信息中指定的端点上。连接标识符（ConnectionId）是一个十六进制字符串，最长 32 字符，大小写无关。MGCP 本地连接属性不受字符集属性控制。

如下为一个 LOCAL 会话描述的例子：

```
v=0
o=- A7453949499 0 LOCAL EPN X35V3+A4/13
s=-
c=LOCAL EPN X35V3+A4/13
t=0 0
a=MGCPlocalcx:FDE234C8
m=audio 0 LOCAL 0
```

注意，MGCP 本地连接属性在会话级指定，在每个端点仅支持 LOCAL 连接的情况下，它可以被省略。

8 基于 UDP 的协议传输

MGCP 在 UDP 上传输。命令发送到 DNS 内为特定端点定义的某个 IP 地址。响应消息回送给命令的源地址（IP 地址和 UDP 端口号），响应不一定来自与命令的发送地址相同的地址。

在没有为端点指定端口时，命令的发送端口为：

- 呼叫代理将命令发送给网关的默认 MGCP 端口 2427
- 网关将命令发给呼叫代理默认的 MGCP 端口 2727

8.1 最多处理一次（at-most-once）

在 UDP 上传送的 MGCP 消息可能丢失。在规定的时间内没有收到响应，命令被重发。多数的 MGCP 命令不是幂等的。例如，建立连接命令被多次执行后，网关的状态将变成未知。因此传送过程必须提供最多处理一次功能。

MGCP 实体被期望能够记住正在执行的事务序列和最近发出的事务响应序列。收到命令的事务标识符将和最近响应的事务标识符进行比较。如果发现匹配，MGCP 实体将不再执行该命令，而是简单地重发响应。否则，再和当前的事务序列相比较（也就是说，先前收到但还没有执行完的事务）。如果发现匹配，MGCP 实体将不执行该事务，但应该发送一个临时响应，证实该事务的接收。

该过程使用一个长定时器，下文用 T-HIST 来表示。此定时器的设定值必须大于一个事务的最大持续时间，要充分估计最大重传次数、重传定时器最大值以及网络传输的最大延迟。其建议值是 30

秒。

响应的拷贝可在以下情况下被删除：发出响应后又过去了 T-HIST 秒；或者是网关（或呼叫代理）通过“ 响应证实 ”参数收到了响应已被接收的证实信息。对于通过该参数证实的事务，网关将在响应发出后的 T-HIST 秒内保存事务标识符的拷贝，用于检测和忽略网络可能产生的重复的事务请求拷贝。

8.2 三次握手机制（Three ways handshake）

处理标识符是 0 到 999999999（包括它们）之间的一个整数。呼叫代理可以决定为其管理的每个网关分配一个指定的处理标识符的号码区间，或者对属于随以分组的所有网关使用一样的号码段。呼叫代理可以决定在管理一个大型网关的多个独立进程间进行负载分担。这些进程必须共享事务号码区间。有多种可能实施这样的共享，比如，进行处理标识符的集中分配，或者为不同进程预先分配不重叠的标识符范围。但必须保证所有源自一个逻辑呼叫代理的事务均分配一个唯一的事务标识符。网关只能通过事务标识符判断重复的事务。

响应确认属性在任何命令里均可发现。它传送了一组收到的最终响应的“ 确认的事务标识符的范围 ”（临时响应不要确认）。特定的相应不应该在两个分开的消息中确认。

如果在响应证实消息（命令或响应）中包含“ 证实的事务标识符范围 ”，MGCP 实体可以选择删除对事务标识符落在该证实范围内的事务的响应的拷贝（但不是事务标识符）。后续的来自该实体的命令，如果其事务标识符落在该范围内，就被丢弃；且在小于 T-HIST 秒前，会发送响应。

当证实响应时，实体必须非常小心。尤其，响应应该只在响应证实发送到的实体与相应命令发送到的实体相同时才被证实。

同样地，实体不应该盲目地对接收特定响应的响应证实。但是，当响应证实的发送实体与产生该响应的命令的实体相同时，接收对特定响应的响应证实被认为是安全的。

应该注意，在命令中（与伴随临时响应的响应证实不同）使用响应证实是可选的。使用它的好处是减少整体内存消耗。但是，为了避免大量的消息，实施时不应该产生大量的响应证实。一个策略是以基于每个端点来管理对命令的响应。一个端点的命令可以证实该端点上的以前命令的响应。对带有通配符端点名的命令的响应可以有选择地证实（考虑消息大小）或者干脆不证实（除非响应显式要求了响应证实）。必须注意，不要两次证实相同的响应，也不要证实大于 T-HIST 秒的响应。

如果自从一个实体发送最后响应给另一个实体以后，已经过了 T-HIST 秒，或者一个实体重新开始操作时，“ 证实的事务标识符范围 ”不应该再使用。这时，命令将不进行事务标识符的检查而直接被接受和处理。

带有“ 响应证实属性 ”的命令可能在传送中失序。应该保持对最近收到的命令的“ 证实处理 ID 范围 ”进行合并。

8.3 重传定时器（Retransmission timer）

请求实体有责任为所有未处理完的命令提供合适的超时控制，并在超时时对命令进行重传。此外，当重复的命令证实失败时，请求实体有责任寻找备份服务，并清除现有的或挂起的关联。

本规范故意避免为重发定时器指定任何值。通常，这些值与网络有关。重传定时器一般通过测量在发出命令与收到第一个响应间所花的时间来估计。至少，应该执行一个包括指数补偿算法在内

的重传策略。一种可能是使用 TCP-IP 执行的算法，该算法用到两个变量：

- 平均证实时延 AAD，通过对观察到的延迟进行指数平滑平均来估计。
- 平均偏差 ADEV，通过对观察时延和当前平均值的差值的绝对值进行指数平滑平均来估计。

在 TCP 里的重传定时器 RTO 被设置为平均延迟值加上 N 倍平均偏差之和，此处 N 为一个常量。在 MGCP 中，定时器的最大值是有限制的，以此保证网关不会在 T-HIST 秒后收到重复的分组。建议 RTO 最大值（RTO-MAX）为 4 秒。实施者也要考虑限制该定时器的最小值。

在任何重传后，MGCP 实体将执行以下操作：

- 将估计的对该事务的证实时延翻倍，T-DELAY。
- 计算一个均衡分布在 0.5T-DELAY 到 T-DELAY 间的随机值。
- 设置重传定时器值为此随机值加上 N 倍平均偏差之和与 RTO-MAX 中的较小者。

该过程有两个作用。因为它包含了一个指数递增部分，它将在拥塞时自动降低消息流量。因为它包含了一个随机部分，将会破坏同一个外部事件触发的通知间潜在的同步。

注意，估计因子 AAD 和 ADEV 不应该对需要重传的事务进行升级。在一个成功的重传后的第一个新的传送应该使用上一次重传的 RTO。如果该传送成功且没有任何重传，AAD 和 ADEV 估计因子被升级，且 RTO 又像平常一样来确定。

8.4 分段和重组

在 UDP 上传送的 MGCP 消息依靠 IP 来对大的数据报进行分段和重组。IP 数据报大小的最大理论值为 65535 字节。除去 20 字节的 IP 头和 8 字节的 UDP 头，理论上最多剩下 65507 字节可以用于在 UDP 上传送的 MGCP 消息。

然而，IP 不要求主机接收大于 576 字节的 IP 数据报，这将导致一个不能接受的很小的 MGCP 消息尺寸。因此，MGCP 要求实施中支持的 MGCP 数据报大小直到至少 4000 字节，这就要求支持相应的 IP 分段和重组功能。注意，4000 字节的限制使用与 MGCP 层。低层开销将会要求支持

大于该字节数的 IP 数据报：UDP 和 IP 开销至少为 28 字节，而且，使用 IPSec 还会带来额外的开销。

应该注意，上述规定适用于呼叫代理以及端点。呼叫代理可以通过审查端点来决定是否它们支持比上述规定更大的 MGCP 数据报。端点目前并没有相似的能力来确定是否呼叫代理支持更大的 MGCP 数据报。

8.5 捎带传送

有时候，呼叫代理要同时发送多个消息给同一个网关，反之亦然。当多个 MGCP 消息要在同一个数据报中发送时，消息之间用一个只包含一个圆点的行分开，见下例：

```
200 2005 OK
.
DLCX 1244 card23/21@tgw-7.example.net MGCP 1.0
C: A3C47F21456789F0
```

I: FDE234C8

被捎带传送的消息必须被视为好象它们在几个分开的数据报中被收到一样来处理。数据报中的每个消息都要被按顺序处理完，且从第一个消息开始，每一个命令都要被响应。在一个消息的处理中发生了错误，不影响其它的消息处理。每个消息都单独处理。

捎带传送用于达到以下两个目的：

- 保证消息的顺序传送和处理；
- 共同的消息发送结果；

当捎带传送用于保证消息的顺序传送时，实体必须确保顺序传送属性在重传每个消息时依然保有。例如，当多个通知（NTFY）命令使用捎带传送时。

共享消息的发送结果保证了或者所有的消息都发送成功，或者全都不成功。当捎带传送用于保证结果共享时，实体必须保证该属性在重传时依然保有。例如，当收到一个来自以单步（Lockstep）方式操作的端点的通知命令时，呼叫代理或许希望在一个数据报中发送对通知的响应和一个新的通知请求命令，确保这两个消息的接结果共享。

8.6 临时性响应

某些事务的处理时间可能很长。很长的处理时间可能会影响重发过程的定时器。这样可能会导致重发次数紊乱，或定时器变得很长，降低效率。

网关（或呼叫代理）若能预计出一个事务将需要一个长的处理时间，它应该发送一个响应码为 100 的临时响应。规定，需要外部通信（例如，网络资源预留）才能完成的事务，应该发送临时响应。此外，实体在收到重发过来的没有执行完的事务时，应该发送临时响应。

开始建立将要执行地事务的队列的网关（或呼叫代理）可以发送响应码为 101 的临时响应来指示这一点。

纯粹的事务的语法表明，临时响应不应该返回任何不是当前正在执行的事务的其他信息，然而，优化的方法是允许返回一些信息，使得系统中减少延时。

为了减少系统中的延时，推荐在对建立连接的响应码为 100 的临时响应中包含连接标识符和会话描述符。如果修改连接命令需要返回一个会话描述符，该描述符也应该包含在临时响应中。如果事务成功处理完，临时响应中的信息必须在最终响应中重复出现。在成功响应中不重复这些信息或者改变以前提供的任何信息都被认为是一个协议错误。如果事务处理失败，返回一个错误码，以前返回的信息也不再有效。

如果某个端点收到了删除连接命令，当前正在执行的建立连接和修改连接命令必须被取消。在这种情况下，对被取消的命令的最终响应仍应自动被返回（推荐使用错误码 407 – transaction aborted）；如果发现重传了被取消的命令，则要返回对被取消的命令的最终响应。

收到临时响应的 MGCP 实体应该切换到一个该事务的长重复定时器（LONGTRAN-TIMER）。该定时器的主要目的是检测处理失败。它的缺省值为 5 秒，然而，赋值进程可以改变该缺省值。注意，重传仍必须满足本协议的相关要求。因此，LONGTRAN-TIMER 必须小于 T-HIST。实体也不必要让事务永远运行。被实体按超时处理的事务应该返回错误码 406（transaction time-out）。对每一个 T-HIST 定义，最大的事务处理事件都要小于 T-HIST（在时延很低的网络中，它可以用 T-HIST 减去 T-MAX

来近似估计），且最终响应应该在初始命令发送后不多于 T-HIST 秒内收到。不过，实体在放弃接收最终响应前应该等待 2*T-HIST 秒。命令的重传仍应在 T-MAX 秒后停止。如果没有收到响应，则不知道事务的处理结果。如果发送命令的实体时网关，它现在会变为“ 断开 ”，且应该启动“ 断开 ”过程。

当事务处理完后，发送最终响应，同时删除废掉的临时响应。为了确保尽快检测到丢失的最终响应，在对事务的临时响应后发送的最终响应应该被证实。

因此，端点应该只在最终响应中包含一个空的“ 响应证实 ”（ResponseAck）参数。最终响应参数中的“ ResponseAck ”参数的出现会触发一个“ 响应证实 ”（Response Acknowledgement）响应发送到端点。该“ 响应证实 ”响应在响应头中包含它证实的响应的事务标识符。

和命令相对应，“ 响应证实 ”响应的接收受相同的超时和重传策略和过程的支配。也就是说，最终响应的发送者在没有按时收到“ 响应证实 ”响应时，会重传最终响应。“ 响应证实 ”响应无需再证实。

9 状态、出错倒换和纷争状况

呼叫代理必须时刻掌握终端的状态，以便发送正确的呼叫信令，而网关必须确保正确把事件发送给呼叫代理。当网关或呼叫代理重启动时会面临一些特殊状况：当呼叫代理进行“ 出错倒换 ”（failover）时，网关必须重新连接到一个新的呼叫代理上，而当网关脱离服务或重启动时，呼叫代理也必须采取一些特殊的应对措施。

9.1 MGC/CA出错倒换

以下几点对于理解呼叫代理的出错倒换处理机制是非常重要的：

- 呼叫代理用域名（以及可选的端口号）而不是它的 IP 地址来标识，同一域名可以对应多个 IP 地址。
- 在某一时刻，一个端点有且仅有一个呼叫代理与之关联。与端点相关的呼叫代理是通知实体（notified entity）的当前值，通知实体决定网关将把它的命令送往何处。如果通知实体参数值不含端口号，则使用呼叫代理的缺省端口号 2727。
- NotifiedEntity 参数是呼叫代理发送给网关的一个参数，用来设置端点的通知实体参数。
- 端点的“ 通知实体 ”总被设为最后收到的 NotifiedEntity 参数值。如果没有收到明确的 NotifiedEntity 参数，“ 通知实体 ”将设置为事先提供的缺省值。 如果没有预置的缺省值或者 NotifiedEntity 参数为空值（会导致通知实体为空，本规范不提倡这两种做法），“ 通知实体 ”将被设为最后收到的非审计命令的源地址。也就是说，审计命令不会改变“ 通知实体 ”。
- 对命令的响应要发送给该命令的源地址，与当前的通知实体无关。当 Notify 消息需要通过对所收到新命令的响应来捎带传送时，数据报仍要发送给该命令的源地址，与当前的“ 通知实体 ”无关。

“ 通知实体 ”可以被解析为多个网络地址，这使得“ 通知实体 ”能够将呼叫代理体现为具有多个物理接口的系统或（和）由多个物理系统组成的逻辑实体。对于有多个 IP 地址的域名，域名服务器（DNS）解析出来的 IP 地址的顺序是不确定的，因此，呼叫代理的出错倒换也不可能在多个 IP 地址间依照某种顺序进行。（例如，一个网关必须可以向多 IP 呼叫代理的任何一个 IP 地址发送 Notify

消息。），网关将命令发送到当前已与它建立连接的那个接口上，系统效率可能最高。为此，建议网关采取以下机制：

- 如果“ 通知实体 ”有多个 IP 地址，而收到请求的源地址是其中一个地址，那么这个地址应被作为随后命令的优选目的地址。
- 另一方面，如果收到请求的源地址不属于“ 通知实体 ”的多个地址，则网关必须从“ 通知实体 ”的多个地址中选择一个作为随后命令的目的地址。
- 如果网关无法联系到所选的 IP 地址，则必须向其它 IP 地址发起新的尝试（见 9.3）。

如果整个呼叫代理变为不可用，它所管理的端点最终状态将成为“ 断开的 ”(disconnected)。使这些端点恢复连接（connected）的方法是：出错的呼叫代理重新正常工作，或者备用呼叫代理通过发送新的“ 通知实体 ”来接管受影响的端点。

当备用呼叫代理接管一组端点的控制之后，故障呼叫代理通常应该与备用呼叫代理联系并保持同步，以便备用呼叫代理可以将对端点的控制再交还给主呼叫代理。也可以是故障呼叫代理只是简单地变为备用呼叫代理。

注意，本规范不考虑相互独立的呼叫代理之间的倒换冲突解决方案，而考虑的是呼叫代理之间彼此知道在做什么保持互相通信（尽管通过 AuditEndpoint 命令可以到当前的“ 通知实体 ”）。如果不是这种情况，可能发生意想不到的事情。

如前所述，缺省的“ notified entity ”需要被设置，可以包含域名和端口。对于小网关，可以为每个端点进行设置，对于大网关，可以给多个端点甚至整个网关设置一个缺省值。在以上两种情况下，当网关加电后，每个端点必须获得它自己的“ 通知实体 ”参数值，因此为一组端点设置的缺省值必须在网关开始工作前被拷贝到每个端点的“ 通知实体 ”中。如有可能，发送给预置的“ 通知实体 ”的重启动（RestartInProgress）命令最好使用“ all of ”通配符，这样会减少发送的 RestartInProgress 命令的个数。

“ 通知实体 ”的另一个作用是体现了网关和呼叫代理之间的一种关联。“ 通知实体 ”就是建立这种关联的纽带，并且由它来控制网关把命令发往何处。网关收到的命令可能来源于任何呼叫代理，网关加电启动后，在收到来自呼叫代理的 NotificationRequest 命令之前，所发生的永久事件以及 RestartInProgress 命令将被发送到缺省的呼叫代理。一旦呼叫代理发起了一个请求，它就可能会带有 NotifiedEntity 参数，且在呼叫代理与网关之间建立一个新关联。在呼叫过程中“ 通知实体 ”是保持不变的，所以这个关联在收到新的“ 通知实体 ”之前也保持不变。

9.2 与网关通信

在网关中端点总是以一个本地名加上一个域名来命名的，本地名用于标识网关内具体的端点，而域名用来标识端点所在的主机。 网关可以有多个接口用于冗余处理。

在具有选路能力的网关中，域名可以解析成一个网络地址，从网关的任何一个接口都可以内部路由到这个地址；另一方面，域名可以解析成多个网络地址，对应每个接口分配一个地址。在后一种情况下，当一个呼叫代理与网关的一个地址联络失败后，它必须尝试联络其他地址。

9.3 重传、检测关联丢失

MGCP 协议是通过一系列事务（Transactions）机制来实现的，每个事务由一个命令和一个响应

（通常叫做对命令的确认）组成的。通过 UDP 传送的 MGCP 消息容易丢失，在一段时间内如果命令发送方没有收到响应，命令就会被重传。MGCP 实体必须保存一个近期事务的响应队列，该队列保存着过去 T-HIST 秒内实体发送的响应；另外，实体还需要保存一个事务队列，用于存放当前正在处理的事务。

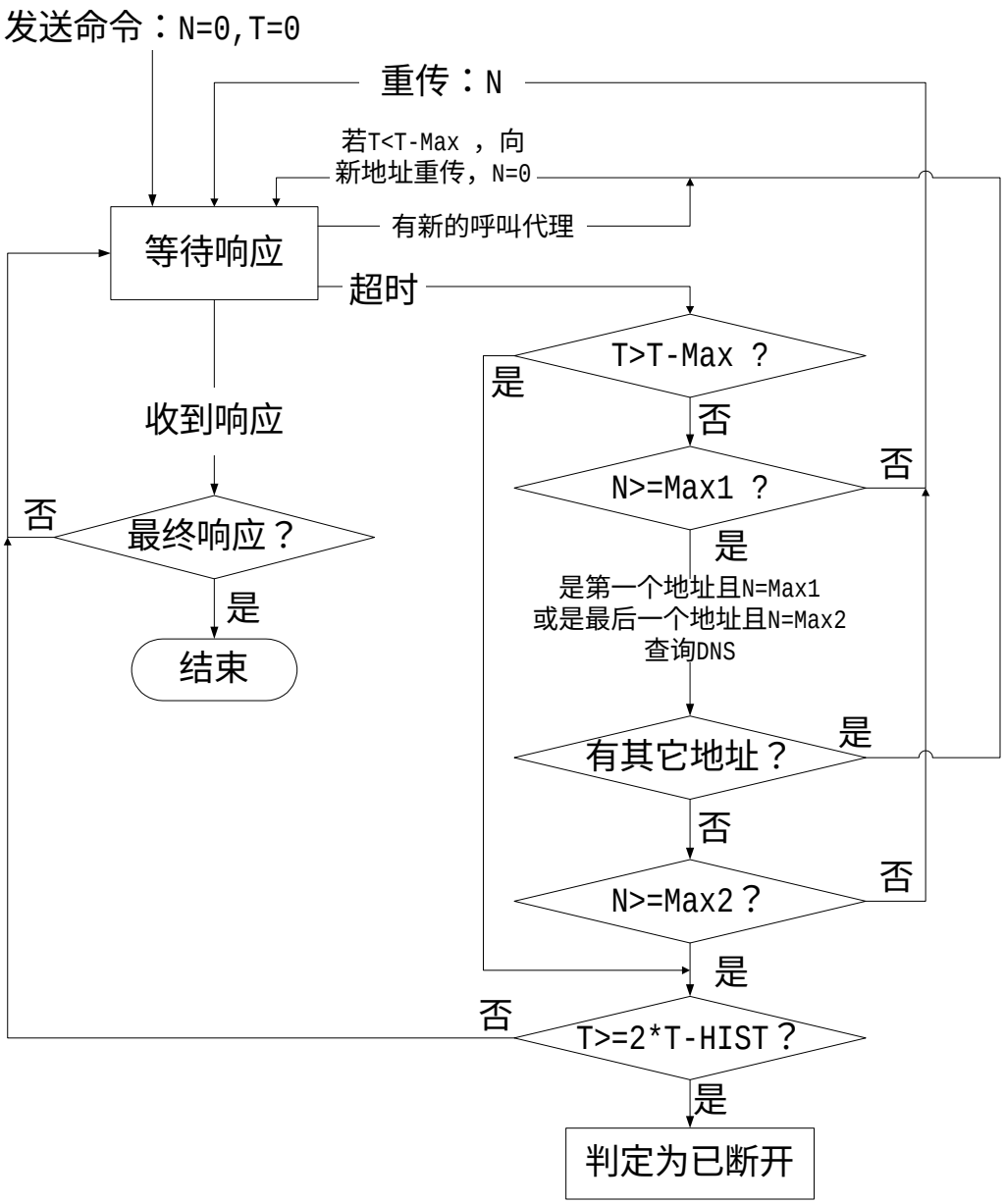
当 MGCP 实体收到命令时，会将命令的事务标识符(TransactionID)和近期响应队列中所有响应的事务标识符进行比较，当发现有匹配的事务标识符时，MGCP 实体只是把响应重发一遍，而不处理该命令；如果没有发现匹配的事务标识符，实体就会进一步将刚收到命令的事务标识符与当前正在处理的事务队列进行比较，如果发现有匹配的事务标识符，实体将发送一个临时响应，而并不处理该刚收到事务，当该事务处理结束后才会发一个最终响应。

重传机制用于防止以下四种可能的错误：

- 传输错误，例如，一个包可能因为线路上的噪音或者队列中阻塞而丢失。
- 组成单元错误，例如，一个与呼叫代理的接口变为不可用。
- 呼叫代理错误，例如，整个呼叫代理变为不可用。
- 出错倒换，呼叫代理因出错而被一个新的呼叫代理透明地接管。

应该可以从历史记录中估算到由于传输错误导致的丢包率。在配置合理的系统中，这个丢包率应该很低，通常小于 1% 。但是当一个呼叫代理或者网关不得不多次重发消息时，就可以判断所发生的故障并不是简单的传输错误。例如，假设丢包率是 1%，那么发生 5 个连续的传输错误的概率应该是千亿分之一，也就是说该故障对一个处理能力在每秒 1000 个事务的呼叫代理来说也只是十几天才会发生一次。（的确，这个过多的重传次数应该是当前丢包率的一个函数）注意，“ 可疑重传次数阈值 ”（下面称为 Max1）通常要比“ 断开重传次数阈值 ”（下面称为 Max2） 低，即，Max2> Max1。

下面是 MGCP 的重传算法的描述图和详细解释。



传统的重传算法只是简单的累计已重传的次数，并且当重传次数达到了较大值（通常是 7 ~11 次）后就判定联接已断开（disconnected）。为了估计未发现的或正在处理的“ 错误倒换 ”的可能性，本协议对传统的算法进行了以下的修改：

- 网关应时刻检查是否出现新的呼叫代理，在出现以下两种情况时，表示有新的呼叫代理：
- 收到一个带有 NotifiedEntity 参数的命令且该参数值指向了一个新的呼叫代理。
- 收到一个指向一个新呼叫代理的重新选路响应。
- 当发现了一个新的呼叫代理，网关必须将未发送的事务重新路由到新呼叫代理，而对无论是来自新呼叫代理还是来自旧呼叫代理的命令的响应都应回复给命令的源地址。
- 在每次重传事务之前，都会检查离发送事务时间是否已经超出 T-MAX，如果超出，重传必须停止。当时间超过了 2*T-HIST 时，端点将被判定已断开联接（disconnected）。

- 当对一个呼叫代理进行重传的次数达到了“ Max1 ”，如果最近（指 5 秒钟或预先定义的一段时间内）该呼叫代理的域名都没有被使用过，那么网关就可以主动向域名服务器发起查询，以便发现与呼叫代理的接口是否已改变。
- 网关可能已经获得了呼叫代理的多个 IP 地址，当向一个 IP 地址的重传的次数大于等于“ Max1 ”而小于“ Max2 ”，而网关还有未试过的 IP 地址时，网关必须向这些未试过的 IP 地址发起重传。另外，当收到来自网络的明确提示，例如 ICMP 中关于网络、主机、协议或端口不可到达的错误报告，也将会导致网关向其他呼叫代理发起新的连接尝试（这也是出于安全的考虑）。
- 当没有接口地址可以再试，并且已对当前地址重传的次数达到 Max2 时，除非最近（指在 5 秒内或其他设定值）解析过域名或者现在正在解析域名，网关都应该再次向域名服务器查询是否有其他可用接口。如果仍然没有接口可以尝试，网关将被断掉而且网关必须开始“ 断开 ”处理过程（见 9.4.7）。

为了自动适应网络负载，MGCP 定义了一系列以指数规律增加的计时器。当初始的计时器是 200 毫秒时，第五次重传的丢失将在 6 秒钟后被发现，这是足以发现出错倒换的可能的等待延迟时间。在延迟等待后重发还应该继续下去，这不仅仅是为了克服瞬时的传输问题，也是为了给执行出错倒换争取更多的时间---总共 30 秒的延迟应该是可以接受的。

最大重传延迟是有限的，在任何一次重传之前，都会检查离发送事务时间是否已经超出 T-MAX，如果超出，重传必须停止。当时间超过了 2*T-HIST 时，端点将被判定已断开联接。T-MAX 的值与 T-HIST 的值有关，T-HIST 的值必须大于等于 T-MAX 值与网络最大延迟的总和。

T-MAX 的缺省值是 20 秒，而假设网络最大延迟是 10 秒，那么对已经发送了的响应就会被保存在响应队列里至少 30 秒。重要的是发送方和接受方必须协商达成这些值。

Max1 的缺省值是 5 次重发而 Max2 的缺省值是 7 次重发。这两个值都可以通过配置程序进行修改，配置程序还必须可以禁止域名服务器查询 Max1 和 Max2。

9.4 纷争状况

MGCP 是根据“ 隔离列表 ”（ “ quarantine list ”）的概念和对同步混乱的明确监测来处理纷争状况（race condition），譬如，由于对一个端点的双占用（也称同抢）造成的摘/挂机状态混乱就是一种纷争的例子。

MGCP 的传输机制并不能保证命令和响应的顺序发送和接收，但这样就会造成纷争的发生，如果呼叫代理处理得当就可以避免这种情况的发生（注意，有些纷争情况是分布式系统固有的，即使命令是严格按照顺序发送也会有纷争发生 ）。

在某些情况下，许多网关会在同一时刻进行重启，例如在某一地区停电或者传输因为地震、冰暴而中断后，当再恢复时，许多网关将同时发送 RestartInProgress 命令，这将导致呼叫代理运行非常不稳定。

9.4.1隔离列表

用 MGCP 协议控制的网关将收到“ 通知请求 ”（NotificationRequest）命令，要求网关监视一系列事件的发生。协议中决定这些事件的元素有“ Requested Events ”列表、“ Digit Map ”、

“ Quarantine Handling ”参数和“ Detect Events ” 列表。

在端点初始化时，“ Requested Events ”列表只包括该端点的永久事件，这时的号码表（Digit Map）一般是空的。此时，端点可以用保留的 RequestIdentifier 参数值“ 0 ”通过 NotificationRequest 消息来报告一个永久事件，例如，摘机事件。事先存在的摘机状况也必定会导致摘机事件的产生。

在端点收到 NotificationRequest 命令后，网关就开始监视端点的命令中所列事件，同时包括永久事件。

事件一发生就会被检测到，接下来的处理取决于 “ Requested Events ”中与事件相关的“ action ”参数或号码表（Digit Map）。那些被定义为“ 累计 ”或“ 根据号码表累计 ”的事件将被累积在一个事件列表中，并且 “ 根据号码表累计 ”的事件还会另外被累积到“ 当前已拨字符串 ” 中。这种累积过程会一直持续直到收到了可以触发 Notify 命令的事件，Notify 命令将被送给当前的“ 通知实体 ”（notified entity）。

这时，网关发送 Notify 命令并将该端点置为“ 通知 ”状态。只要端点处于通知状态，需要在该端点上监视的事件都会被放进先入先出的“ 隔离 ”（quarantine）缓冲区里等待处理。这些事件从某种意义上来说是被“ 隔离 ”了，所有在 RequestedEvents 参数中和最近收到的 DetectEvents 参数中所列出的事件，或者仅在 RequestedEvents 参数中要求的事件都应该被隔离，而不考虑与该事件对应的处理（action）是什么。 在这里永久事件被认为是隐含在 RequestedEvents 中的。当一个端点的隔离缓冲区容量达到上限时，就应该产生一个隔离缓冲区溢出的事件（见 RFC3435 附录 B。如果支持隔离缓冲区溢出事件，端点必须保证该事件本身可以被放在隔离缓冲区中），那些溢出的事件将被丢弃。

当收到 Notify 命令的响应后（无论成功还是失败），端点将退出“ 通知状态 ”。Notify 命令可以在“ 通知状态 ”下进行重传。如果端点在通知状态断开了连接，给 Notify 命令的响应将永远无法被端点收到。虽然此刻的端点仍然处于“ 通知状态 ”，该 Notify 命令将被丢弃并且该端点也不再被认为是在等待 Notify 的响应。只有完成了断开连接处理（见 9.4.7）后，端点才可以退出“ 通知状态 ”。端点退出“ 通知状态 ”时，它将清空观察到的事件 （ObservedEvents）列表和“ 当前已拨字符串 ”。

接下来网关在触发 NotificationRequest 命令时的处理取决于 QuarantineHandling 参数值：

如果呼叫代理已经规定对于每个 NotificationRequest 命令，最多只能有一个 Notify 命令作为响应，那么网关只需要简单地将事件累积到隔离缓冲区中，直到它收到了下一个 NotificationRequest 命令。

如果网关被授权可以针对一个 NotificationRequest 命令发送多个连续的 Notify 命令，它将进行下面的处理。当网关退出“ 通知状态 ”时，它将清空 ObservedEvents 列表和“ 当前已拨字符串 ”并开始处理隔离事件列表，其中包含已经收到的请求事件（requested event）列表和号码表中的事件。在处理这些隔离事件时，网关可能会遇到一个可以触发 Notify 命令的事件，当遇到这种事件时，网关可以采取以下两种措施之一进行处理：

- 网关可以立即发送 Notify 命令来报告在 ObservedEvents 列表中累积的触发事件以前的事件（包括触发事件），而触发事件之后那些事件仍被留在隔离缓冲区里。
- 网关也可以试着清空触发事件之前的隔离缓冲区，并且用一个 Notify 命令来报告多组事件（放在单个观察事件（ObservedEvents）列表中），可能是一些拨号字符串。在每个触发

事件后，需要清空“ 当前已拨字符串 ”。在最后一个触发事件之后的所有事件依然被保留在隔离缓冲区内。

如果网关发送了一个 Notify 命令，端点将会重新进入“ 通知状态 ”并且在收到响应以前会一直保持该状态。如果网关并没有发现可触发 Notify 命令的隔离事件，它就把端点置成常规状态（normal state）。新检测到的事件会用 Notify 命令上报就像已经收到了 NotificationRequest 命令一样。

网关可以在任意时刻收到针对某端点的新 Notification Request 命令，即使在该端点断开时。除了当前的 ObservedEvents 列表保持不变而不是被再次处理外，激活嵌入的通知请求被当作收到一个新的 NotificationRequest 命令一样来处理。当在通知状态收到了新的通知请求时，网关应该确保呼叫代理在收到新的 Notify 之前先处理上一个 Notify 命令（注意，由于联接断开而丢失的 Nofify 是再也不可能被处理的）。网关是利用协议的“ 捎带传送 ”的功能来实现这些的， 几个消息被放在一个包里被送到当前的“ 通知实体 ”。按以下步骤操作：

a) 网关对新收到的 NotificationRequest 发送一个响应。

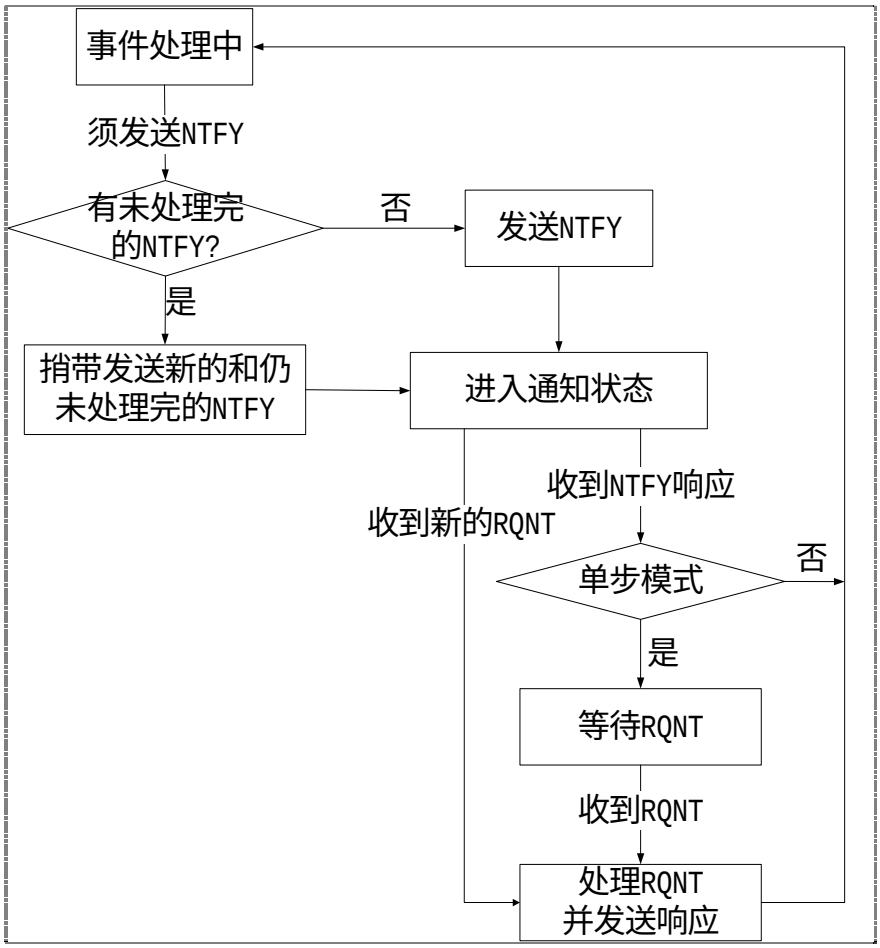
b) 端点将跳出“ 通知状态 ”而不去等待未处理完的 Notify 命令的确认消息。

c) 未收到确认的 Notify 命令的副本将被保留。如果超时未收到确认，该 Notify 将被重发。

d) 如果网关在未收到已发送 Notify 命令的确认消息以前不得不又要发送一个新的 Notify 命令，它将把那些仍未确认的 Notify 和新的 Notify 放在一个包里发送到当前的“ 通知实体 ”。（这些 Notify 消息是按照发送的时间顺序排列在发送包里的）

e) 那些无法在同一个数据包中传送多个消息的网关必须保证了两个或多个 Notify 消息按顺序发送。端点将保持在“ 通知状态 ”直到收到最后一个 Notify 的确认消息。

下图说明了以上的过程：



网关也可以利用“捎带传送”的功能，在给新的 NotificationRequest 发送成功响应之前，再次发送还没有得到确认的 Notify。这实际上是 RFC2705 中的要求，而这样做会比较复杂，且是否带来好处也是令人怀疑的。 特别是，RFC2705 中的机制并不能保证 Notify 和 NotificationRequests 的响应可以按顺序发送，因此呼叫代理必需总是要处理无序消息的问题。将这种处理变为可选的就可以后向兼容而且会大大降低复杂性。

在收到 NotificationRequests 命令后，请求事件列表和号码表（如果提供了新的）将被更新为最新收到的参数，并且拨号字符串也将被清空。此外，如果 NotificationRequests 是在“通知状态”下收到的，那么观察到的事件（ObservedEvents）列表也将被清空。后续的处理取决于 QuarantineHandling 参数的取值，QuarantineHandling 参数可以确定那些被隔离事件（和此时应该为空的 ObservedEvents 列表）是否应该被丢弃，在什么情况下将被丢弃。如果 QuarantineHandling 参数指出需要处理被隔离的（以及观察到的）事件，网关就会利用最新收到的请求事件和号码表（如果有的话），开始处理这些事件。在处理时，网关可能会遇到一个触发 Notify 命令的事件，这时网关将立即发送 Notify 命令，以报告已收到累积在“观察到的事件列表”中的所有事件，这些事件包括在触发事件以前收到的事件和触发事件。未处理的事件将被留在隔离缓冲区中，随后网关进入“通知状态”。

当网关还在为一个通知请求收集事件并且还没有收到触发事件时，端点还不处于“通知状态”，这时它可能会收到一个新的通知请求。在这种情况下，还没有被通告的事件也象被隔离事件一样，处理也是由 QuarantineHandling 参数决定的。

- 如果 QuarantineHandling 参数明确指出被隔离的事件应该被忽略，那么观察到的事件列表

将被简单地重置。

- 如果 QuarantineHandling 参数明确指出被隔离事件应该被处理，那么观察到的事件列表中的事件将被转移到被隔离事件列表中，观察到的事件列表将被重置，而被隔离事件列表中的事件将被处理。

用单步（lockstep）方式管理端点的呼叫代理应该把给一个成功的 Notify 消息提供的响应和一个新的 NotificationRequest 命令利用捎带传送的机制放在一个数据包里发送出去。

9.4.2显式的检测

多个端点状态的一个关键因素在于摘机挂机的时机。纷争状况将会发生在以下时候：在呼叫代理请求网关检测摘机事件之前用户摘机，在请求网关检测挂机事件之前用户挂机。（即电路交换中典型的“双占用”现象）。

为了避免纷争状况的发生，网关必须在给 NotificationRequest 确认消息之前检查端点的状态，当发生以下情况时，网关应该返回错误：

1) 当电话已经被摘机，而要求网关报告“摘机”事件（返回错误代码 401-话机已处于摘机状态）

2) 当电话已经被挂机，而要求网关报告“挂机”或“拍插簧”事件（返回错误代码 402-话机已处于挂机状态）

此外，个别的信号定义可以指明一个信号只有在某些条件下才有效，例如，只有在主叫摘机后才有可能产生振铃音。如果一个信号有这样的先决条件，那么网关在发现该先决条件没有被满足的时候就必须返回错误。

注意，先决条件的检测是在收到通知请求后进行的，而真正能够引起该条件变化的事件可能已经被报告了或者已经被忽略了，或者已被隔离。

网关的其他状态变量，例如请求事件（RequestEvents）列表或要求发送的信号(requested signals)列表在每次成功地处理 NotificationRequest 后都会被最新的列表参数替换掉，这样做可以避免呼叫代理和网关之间的差异太大。

当 NotificationRequest 处理失败的时候，无论它是否被包含在处理连接（connection）的命令中，网关必须象没有收到过该命令一样照常处理。与其它的事务一样，NotificationRequest 必须作为一个基本的事务被处理，因此，任何由于该命令导致的变化必须被返回。

另外一种纷争状况可能发生在网关在发送了一个 Notify 后很短的时间内又收到了 NotificationRequest 的时候。RequestIdentifier 是用来关联 NotificationRequest 和 Notify 命令的，因此也使呼叫代理可以判断 Notify 命令是在网关收到 NotificationRequest 之前还是之后产生的。这对于避免单个模式的死锁现象是非常重要的。

9.4.3事务的语义

随着事务处理完成时间的逐渐增加（如由外部资源预留引起的），事务化的语义的准确定义就显得越来越重要。尤其是纷争情况，因为它与摘挂机状态相关，所以更要仔细定义。

需要考虑的一个重要问题是：从一个事务开始到结束（无论成功或失败）的这段时间内，前提

条件（如摘挂机状态）可能会改变。总之，成功的处理一个事务要依靠一个或多个前提条件，而这些前提条件可能会在事务开始到结束的间隔中改变。

一个最简单的语义是要求在事务处理的整个过程中都要满足所有的前提条件。一旦某个前提条件在事务处理结束之前变得无法满足，那么该事务处理就失败了。

以要求检测“ 摘机 ”事件的事务处理为例，在该事务刚发起时，电话处于“ 挂机 ”状态，这样是满足检查摘机事件的前提条件的。如果在这个事务处理结束前话机状态变成了“ 已摘机 ”，那么就不再满足检查摘机事件的前提条件，因此该事务处理就立刻以失败告终。

最后，还要考虑旧事务对端点的处理结束、同时新事务开始生效的那个时刻。例如，假设事务 T1 已经被成功地执行并且正按照 T1 的要求进行事件处理时，收到了一个新的事务 T2（例如，该事务的命令是一个嵌套着 NotificationRequest 的 CreateConnection）。由于还不知道 T2 的处理是否能够成功，我们在没有得到 T2 处理的结果之前，无法开始 T2 所要求的事件处理。虽然我们可以停下来所有的事件处理，等待 T2 处理的结果，但这样将降低系统响应的灵活性，因此不应该这样做。

相反地，当收到一个新的事务 Ty 时，Ty 将依照旧的事务 Tx 来修改自身的事件处理，这样的处理将一直保持到返回成功的 Ty 处理结果。如果 Ty 处理失败，会继续按照 Tx 来处理。所有的由于 Ty 引起的变化在逻辑上都将在 Ty 返回后才生效，因此，如果在 Ty 返回时端点正处于通知状态，而且 Ty 中包括一个 NotificationRequest，那么在 Ty 返回时端点才会跳出通知状态。注意，这与在 Ty 发起时，端点是否已在通知状态没有关系。例如，可以在从 Ty 开始到结束的这段时间内，按照 Tx 的处理需求生成一个 Notify 命令 。如果 Ty 的返回将导致端点进入通知状态，那么，需要一个新的 NotificationRequest（事务 Tz）才会退出通知状态。这样就遵循了事务是按照因果顺序被处理的事实。

另一个相关的问题是通配值的使用，尤其是用于一个或多个端点的“ all of ”通配值。在请求一个命令，而且端点标识符合多个端点时，仍然应该遵守事务语义的定义。此时，命令一定是对所有的端点都成功或者都失败。在这个命令之后总会发送一个响应。

9.4.4命令的顺序以及顺序混乱的处理

MGCP 协议并不要求下层的传输协议来保证给网关或端点的命令能够按顺序传送。这个特性往往能最大限度的保证端点能够及时发起动作，但它也存在一些缺点，例如：

- Notify 命令经过延迟到达呼叫代理的时候，呼叫代理可能已经发送了一个新的 NotificationRequest 命令。
- 当一个新的 NotificationRequest 在收到上一个 NotificationRequest 的确认之前被送出，那么就无法保证哪一个将被先收到并处理。

呼叫代理要想保证对端点的稳定操作必须遵循以下原则：

1) 当一个网关处理多个端点的时候，给不同端点的命令可以同时被发送，例如，可以遵循每个端点可以由一个单独的进程或线程来控制的模式。

2) 当一个端点建立了多个连接时，给不同连接的命令可以同时被发送。

3) 对于一个连接，通常只有一个未处理完的命令（CRCX 或 MDCX 命令）而 DeleteConnection 命令却可以在任何时刻发送。结果，网关有时候就会收到针对一个已经被删掉的连接的修改连接命令（ModifyConnection），这样的命令处理就会失败，而且必须返回一个错误代码（建

议使用错误代码 515---不正确的连接标识）

4)对于一个端点，在某一时刻通常只有一个未处理完的 NotificationRequest 命令。必须使用 RequestId 参数来关联一对 NotificationRequest 命令（可以触发 Notify 命令）和 Notify 命令。

5)某些情况下，将对一组端点进行操作的 DeleteConnection 命令（该命令带有隐式或显式的通配值）可能会在一个 CreateConnection 命令还未得到确认时就要被发送。呼叫代理必须一个个地删掉那些此刻仍在创建中的连接。在这个带有通配值的 DeleteConnection 命令得到确认之前，仍然使用该通配值对这一组端点进行操作的 CreateConnection 命令是不可以被发送的。

6)当命令之间相互嵌套传送时，必须坚持所有命令的顺序要求，例如当一个 NotificationRequest 命令嵌套在 CreateConnection 命令中发送时，必须同时兼顾 NotificationRequest 命令和 CreateConnection 命令的顺序要求。

7)AuditEndpoint 和 AuditConnection 命令并不受任何顺序要求的约束。

8)RestartInProgress 命令一定是端点在重启过程中发送的第一条命令。其它任何命令或非重启命令的响应除了审计命令的响应，都必须在 RestartInProgress 命令之后被发送（可以使用捎带传送功能）。

9)当多个消息在一个包中捎带传送时，消息的处理总是按顺序进行的。

10) 对于一个端点，在某一时刻应该仅有一个待确认的 EndpointConfiguration 命令。

网关不许假设呼叫代理遵守或不遵守以上原则。因此，无论命令是否遵循以上原则，网关都必须对这些命令做出响应。为了保证操作的一致性，当以上的原则没有被遵守时，网关应该进行如下处理：

- 当网关正在处理命令（ModifyConnection、NotificationRequest、或 EndpointConfiguration 命令）且处理还没有结束时，在新的事务中又收到了同样的命令，这样网关应该给前面的命令处理返回错误，这也包括在一个命令中嵌套一个或多个命令的情况。建议使用 407（事务中断）作为错误代码。
- 当网关在处理 CreateConnection 命令的时候又收到 ModifyConnection 时，ModifyConnection 命令就应该被拒绝，建议返回错误代码 400（瞬时错误）。注意，这种情况会造成呼叫代理的程序错误。
- 当网关在处理 CreateConnection 或 ModifyConnection 命令的时候，收到了 DeleteConnection 命令，那么未结束的命令处理必须被中断。建议返回错误代码 407（事务中断）。

注意，当因为收到新的命令需要中断旧命令时，无论新命令的处理是成功还是失败，旧命令都应该被中止。例如，在处理 ModifyConnection 时，收到 DeleteConnection 命令后，由于内嵌的 NotificationRequest 导致了 DeleteConnection 命令处理失败，尽管这样，正在进行的 ModifyConnection 命令的处理也必须中止。

9.4.5端点服务状态

正如先前提到的，已配置的端点可能正在服务中或者服务已中断。端点的实际服务状态可以体

现在 RestartInProgress 命令中的 RestartMethod 和 RestartDelay 参数上，或者也可以通过 AuditEndpoint 命令得到。

服务状态影响了端点处理命令的方式。当一个端点正在服务中时，它会处理所有收到的命令，但是当一个端点的服务已中断，它就会拒绝所有的非审计命令（non-auditing commands）而只在可能的条件下处理审计命令。任何由于端点服务中断导致的命令遭拒绝的情况都应该返回错误代码 501（端点未准备好/连接中断）。

除非已经另有声明，否则命令中带有通配符“ any of ”的端点标识就是指那些还在服务中的端点，而带有通配符“ all of ”的端点标识是指所有端点，与它们的服务状态无关。

下表说明了以上的关系，表中描述了当前的服务状态和网关处理命令的关系。最后一列括号中的是返回给审计命令的响应中的 RestartMethod 参数取值：

Restart Method	Restart Delay	2xx received?	Service-state	Response to new command
graceful	Zero	Yes/No	In	non audit: 2xx audit: 2xx (graceful)
graceful	non-zero	Yes/No	In*	non audit: 2xx audit: 2xx (graceful)
forced	N/ A	Yes/No	Out	non audit: 501 audit: 2xx (forced)
restart	Zero	No	In	non-audit: 2xx,405 * audit: 2xx (restart)
restart	Zero	Yes	In	non-audit: 2xx audit: 2xx (restart)
restart	non-zero	No	Out*	non-audit: 501* audit: 2xx (restart)
restart	non-zero	Yes	Out*	non-audit: 501* audit: 2xx (restart)
disconnected	Zero/ non-zero	No	In	non-audit: 2xx audit: 2xx (disconnected)
disconnected	Zero/ non-zero	Yes	In	non-audit: 2xx audit: 2xx (disconnected)
cancel-graceful	N/A	Yes/No	In	non-audit: 2xx audit: 2xx (restart)

注释（*）：

* service-states 一栏中三个标有“ * ”的服务状态将在 RestartDelay 定义的一段时间后改变，到那时，应该发送一个更新过的 RestartInProgress 命令。
* 如果端点在重启过程还没有结束时要对非审计命令返回 2xx，同时应该返回 405 错误代码。
* 在端点发送的 RestartInProgress 中，如果参数 Restart Method 为“ restart ”，且 RestartDelay 参数为非零值，在 RestartDelay 到期前收到非审计命令则返回 2xx 响应，同时应该返回 501 错误代码。

9.4.6预防重启雪崩

假设大量的网关同时加电重启动时都发送 RestartInProgress，那么由于大量的 RestartInProgress 同时到达，呼叫代理将很有可能会崩溃，从而导致在业务重启动期间引起消息丢失和网络拥塞。为了避免呼叫代理发生重启崩溃，本协议要求采用以下措施：

1) 当加电重启动时，网关必须启动一个重启定时器，并将该定时器值初始化为位于 0 和最大等待时延 MWD（Maximum Waiting Delay）之间的一个随机值。当多个网关使用相同的重启定时器值生成算法时，应避免它们之间重启动定时器随机值生成的同步。

2) 网关必须等待以下三种情况之一发生：重启定时器超时，或收到一个来自呼叫代理的消息，或检测某用户事件，例如驻地网关检测到用户摘机事件。

3) 当以上三种情况之一发生，网关就必须开始重启动过程。

重启过程仅要求端点保证呼叫代理能从该端点收到对第一个非审计命令的非重启响应（non-restart response），非重启响应是指除代码 405、 501 和 520 之外的响应。允许端点采用捎带传送来完成此任务。还在服务中的端点会以“ restart ”作为 RestartMethod 参数值，而已中断服务的端点会以“ forced ” 作为该参数的值。因端点还没有结束重启动而被拒绝的命令，应该返回错误代码 405（端点正在重启）。

一旦收到呼叫代理成功的应答消息，重启过程就结束了。如果收到了错误的响应，后续的处理要取决于错误代码：

- 如果错误代码是瞬时错误（4xx），那么必须用新的事务再次启动重启过程。
- 如果错误代码是 521，表明端点已被重新路由，必须用新的事务再次启动重启过程。错误代码是 521 的响应中一定包含 NotifiedEntity 参数，该参数的值就是再次发起重启的“ 通知实体 ”。如果响应中没有 NotifiedEntity，响应将会被当作一个其它的永久错误。
- 如果错误代码是其他永久错误（5xx），并且端点无法纠正该错误，除非另有规定，否则端点就不自动发起重启过程（直到重新加电重启）。但如果收到了给这个端点的命令，该端点就必须再次启动重启过程。

注意，如果 RestartInProgress 命令是与一个对收到的命令的响应（R）一起捎带传送的，那么 RestartInProgress 命令的重传将不要求重传 R。然而，当端点重启时，重传响应 R 则要求 RestartInProgress 捎带传送，以保证他们的顺序发送处理。

在执行重启的过程中，网关可能会进入“ 断开 ”状态，这样就必须执行 4.4.7 所描述的断开处理，除非在该过程中，网关发送了重启方式为“ restart ”而不是“ disconnected ”的 RestartInProgress 命令。

每个端点都有一个缺省的呼叫代理，即“ 通知实体 ”，当发起重启时将消息发送给该缺省的呼叫代理。当在一个网关上的端点由多个呼叫代理管理时，对于每一组由一个呼叫代理管理的端点都必须执行以上的重启过程。网关必须充分利用通配值来使在一个网关中由一个呼叫代理管理的多个端点同时重启时发送 RestartInProgress 的数目减到最小。注意，在重启时，端点也可以从服务断开状态开始，随着网关的重启变成在服务中状态。一个网关可能会为他所有的端点首先发送一个重启方法为“ forced ” 的 RestartInProgress 命令，然后再为那些已经投入服务的端点发送重启方法为“ restart ”的 RestartInProgress 命令。当然，网关也可以只为那些在服务中的端点发送重启方法为“ restart ”的 RestartInProgress 命令而为那些中断的端点发送重启方法为“ forced ”的

RestartInProgress 命令。此时为了减少发送的消息数量，仍然需要使用通配符。

MWD 是网关中与网关类型相关的配置参数。驻地网关（RG）的等待时延可以用以下因素来计算。

通常呼叫代理的规模应满足峰值小时话务负荷的需求。在负荷峰值期间，平均有 10%的模拟电话线用户处于忙状态，且平均呼叫时长为 3min。通常，呼叫代理与端点之间完成一次呼叫需要 5-6 个事务（Transaction）。简单估算，平均每三十分钟 MGC 需要为每个端点处理 5-6 个 Transaction，或者说平均每 5-6 分钟为每个端点处理一个 Transaction。因此，建议 RG 的 MWD 值为 10-12min. 当 MG 未明确定义 MWD 值时，MWD 应选取缺省值 600s（10 min）。

对于中继媒体网关（TG）或商务媒体网关（Business Gateway）而言，MWD 值应该更小，因为这些网关需要处理的端点数量更多，并且在忙时峰值阶段，端点的利用率也要大于 10%，通常为 60%。因此，在忙时，MGC 的处理能力将按每分钟为每个端点处理一个 Transaction。因此，TG 中每条中继线的 MWD 值应比 RG 中模拟电话线的 MWD 值小 6 倍，并且与正在重启动的端点数量成反比。例如，处理 T 1 中继线的 TG 的 MWD 值被设置为 2.5s，而处理 T3 中继线的 MWD 值则被设置为 60ms。

9.4.7断开的端点（Disconnected Endpoints）

除了重启处理过程，网关还有 “ 断开 ” 处理过程，当端点变成 “ 已断开 ” 状态时，就必须发起该过程。注意，端点只在试图与呼叫代理通信时才会变成 “ 已断开 ” 状态。当成为 “ 已断开 ” 时，端点必须按照以下步骤来处理：

1) 端点需启动一个 “ 断开 ” 定时器，该定时器值初始化为位于 1 和一个事先设定的 “ 断开 ” 初始等待时延 Tdinit（disconnected initial waiting delay）)之间的一个随机值，例如 15 秒。当多个端点和网关使用相同的断开定时器值生成算法时，应避免它们生成同步的断开定时器随机值。

2) 网关必须等待以下三种情况之一发生：断开定时器超时，或收到一个来自呼叫代理的消息，或检测某用户事件，例如驻地网关检测到用户摘机事件。

3) 如果以上三种情况之一发生时，网关就为该端点启动 “ 断开 ” 处理过程。如果该过程已经开始，就应该用新的过程来替换旧的过程。另外，从端点被断开或端点为了限制 “ 断开 ” 处理过程执行的频率自己结束上一次 “ 断开 ” 处理过程到用户事件被检测到这段时间里，预置的 “ 断开 ” 最小等待时延 Tdmin（disconnected minimum waiting delay）必定已经到期。假如 Tdmin 还没有到期，端点将重新执行第 2 步的操作而不会影响已经开始的 “ 断开 ” 处理过程。

4) 如果经过了 “ 断开 ” 过程端点仍然保持断开状态，“ 断开 ” 定时器就将加倍，并且网关将用一个新的 transaction-id 再次执行第 2 步的操作。其中，“ 断开 ” 定时器加倍后仍然要小于一个事先给出的 “ 断开 ” 最大等待时延 Tdmax （disconnected maximum waiting delay），例如 600 秒。

断开处理过程与重启动过程非常相似，只是断开处理过程要求端点用 RestartInProgress 命令告知呼叫代理该端点已经断开了。另外，端点也必须保证对呼叫代理的第一个非审计消息的响应中一定要告知端点已经处于断开状态（除非端点已经退出服务了）。按照以上规定，当收到命令 C 时，端点就会将一个 Restart Method 参数为 “ disconnected ” 的 RestartInProgress 命令与给 C 的响应利用捎带功能一起传送给 C 命令的源地址，这个源地址可能不是当前的 “ 通知实体 ” 。该捎带传送的 RestartInProgress 命令不会被端点自动重发，而仅靠随机与 C 命令的响应一起捎带传送以实

现顺序发送处理。呼叫代理会给这个被捎带 RestartInProgress 命令一个正常响应，但响应可能被丢失。如果丢失，端点会再次发送前面那个 RestartInProgress 命令重新开始“ 断开 ”处理过程，这回端点只将 Restart Method 参数为“ disconnected ”的 RestartInProgress 命令（不捎带对 C 的响应）发送给当前的“ notified entity ”。

当呼叫代理得知端点已经断开，呼叫代理可能会审计该端点或者清除所有与该端点相关的连接。注意这种“ 断开 ”过程会导致重传 RestartInProgress 命令，需要按照重传过程来处理这个重传。经过了断开处理过程后，端点可能还处于“ 断开 ”状态。如果端点在断开期间退出了服务，这时端点应该发送一个 Restart Method 参数为“ forced ”的 RestartInProgress 命令。

断开过程在收到了一个成功响应之后就会结束。对于失败响应的处理与重启过程相似。当收到一个失败响应需要重新启动“ 断开 ”处理过程时，仍然需要考虑限制“ 断开 ”处理过程执行的频率（如上描述）。注意，如果 RestartInProgress 命令与一个在断开过程中收到的命令的响应（R）捎带传送在一个 Transaction 中发送时，该 RestartInProgress 命令的重传并不需要捎带响应 R。当然，当端点断开时，需要重传响应 R 则要求 RestartInProgress 命令与 R 一起捎带传送，这样才能保证它们按顺序传送。

如果一组断开的端点有相同的“ 通知实体 ”那么这组端点可以用一个通配值来命名，网关可以用一个适当的通配值来取代一个端点的标识同时发起对多个端点的“ 断开 ”过程。在这种情况下，使用通配值的 RestartInProgress 命令的 Restart Delay 参数应该是被替代的最早的“ 断开 ”过程中的 Restart Delay。注意，一旦这些端点中的一个或多个端点的“ 通知实体 ”随后变化了和/或不再处于断开状态，那么这个使用通配值的断开过程就无法再进行下去了。必须恢复针对个体的断开处理过程。

断开的端点也有可能想发送一个除 RestartInProgress 以外的命令，但只有在重新恢复与呼叫代理的通信后，这样的消息才可能获得成功响应。于是引发如下疑问：在这期间该怎样处理这样的命令。极端的处理是，端点立即删除该命令，但是当端点虽然还没有结束“ 断开 ”处理过程但实际上已经联系到呼叫代理的时候，这样做就不合适了，例如在收到 NotificationRequest 命令后需要立即返回一个 Notify 命令的情况。为了避免这种情况的发生，在收到一个非审计命令后的 T-MAX 秒内，断开的端点不会盲目地删除一个将要发送的命令。

可以为将要发送的命令开辟一个临时的缓冲区来满足以上的要求。可是端点必须保证以下两点：

- 端点不建立较长的发送命令队列。
- 不可以在重新连接到呼叫代理时，向呼叫代理快速发送过多的命令，这样会使呼叫代理崩溃。

在 T-MAX 秒内把命令缓存，一旦端点再次恢复连接时，限制每个端点所缓存命令的发送频率，特别是使用了通配符时，应采用一个可以接受的发送频率使得每个端点只有一个未处理完的命令。如果在 T-MAX 秒内端点没有被连接而是开始了“ 断开 ”处理过程，端点就可以将那些被缓存的命令与 RestartInProgress 命令一起捎带传送。注意，一旦发送了一个命令，就必须在 T-MAX 秒之后才可以重传该命令。

建议 Tdinit 和 Tdmin 的缺省值是 15 秒，Tdmax 的缺省值是 600 秒。

9.4.8 负载控制（Load Control in General）

前面的各小节描述了 MGCP 协议中处理阻塞和过载的一些机制，即：

- 在 UDP 上的重传机制，该机制在一个端点的基础上根据网络和呼叫代理的阻塞情况进行调整。
- 顺序发送命令的指导方针，该方针限制了同时发送的命令的数目。
- 重启过程，避免了重启雪崩中的高流量。
- 断开处理过程，避免了大量断开端点执行断开过程时的高流量。

尽管有以上措施，一组在相同或不同网关上的端点仍有可能在同一时刻发送一个或多个命令，尽管可以规定呼叫代理应该被限定在任何时刻只为它所服务的每个端点处理一个消息，但在实际应用中却不总是这样。类似地，网关也许也不能在同一时刻为它每个端点只处理一个消息。通常这种问题可以通过基于信任（credit-based）机制或者检测并根据观察到的运行状态自动进行调整。本规范倾向于后者，实现如下：

从理论上假设呼叫代理和网关都维持着一个队列，用于存放所有收到的待处理事务（transactions）。为这个事务队列设定一个高水位标记和一个低水位标记值。一旦该队列的长度大于高水位值，实体就应该发送代码为 101 的临时响应（transaction 正在等待处理），直到队列的长度降为小于等于低水位值。这个原则不仅用于 transaction 的处理也用于重传的 transaction 的处理。如果此时实体无法再处理新的 transaction，那么它就应该返回错误代码 409（处理过载）。

另外，网关应该通过监测从某个呼叫代理返回的对一组端点的响应时间，来调整给该呼叫代理发送新命令的频率。如果观察到的响应时间由原来很平滑突然明显上升至某个阈值，或者网关收到了代码为 101（transaction 正在等待处理）或 409（过载）的响应时，网关应该相应地调整给相关呼叫代理的新命令发送频率。使响应平均时间保持稳定，发送频率调整以及阈值的定义等细节还需要进一步的研究，但这些因素必须是可配置的。

同样地，呼叫代理也应该通过监测观察到的从某个网关返回的一组端点的响应时间来调整给该网关的新命令的发送频率。具体的操作与上面相似。

9.5 心跳机制

由于 MGCP 承载在 UDP 协议上， UDP 是不保证可靠传递的。为了 MGC 和 MG 间通信中断后，双方都能够及时的检测到网络中断，MG 和 MGC 之间应该实现心跳机制。主要有两种方式：

9.5.1只有MGC控制的心跳消息

MGC 可以为 MG 设置一个最大沉默时间，即正常工作 MG 允许未收到 MGC 发送的任何消息的最大时间。 MGC 应该保证向 MG 发送消息的时间间隔不超过最大沉默时间。即使在最大沉默时间内没有任何其它消息，MGC 也必须通过向 MG 发送心跳消息来表明自己还“ 活着 ”。

本规范建议 MGC 用针对端点“ mg ”（见 RFC3435 E.4）的空 AuditEndpoint 命令作为心跳消息，网关对该消息应只回一条响应消息。心跳周期在 MGC 中可以设置，象 TG 这样的大型网关可以设短些，而 IAD 则应该设长些，每个心跳周期 MGC 向 MG 发送一个针对端点“ mg ”的 AuditEndpoint 消息。最大沉默时间设为 8 个心跳周期，当 MG 连续 8 个心跳周期没有从 MGC 收到任何消息时，就将所有 MGC 控制的端点置为“ 断开 ”（disconnected）,然后开始断开处理程序，具体过程参见 9.4.7。

MGC 利用事务请求的重传机制依靠定时器超时来判定 MG 断开，当超过定时器 2*T-HIST 时，受控端点被判定已断开（disconnected）。由于有周期性发送的心跳消息，可以保证 MGC 及时检测到

MG 断开。

9.5.2MGC和MG分别独立控制的心跳消息

MGC 和 MG 互相向对方发送心跳消息，心跳周期由各自独立决定。协议实体利用事务请求的重传机制依靠 2*T-HIST 超时来判定对方实体断开，由于有周期性发送的心跳消息，可以保证协议实体及时检测到对方实体断开。与前一种方式相比，MG 可以更自主地控制发送心跳消息的时机。

MGC 仍采用针对端点“ mg ”的空 AuditEndpoint 命令作为心跳消息。

MG 应采用针对端点“ mg ”的 Notify 命令作为 MG 心跳消息，由于 IETF 没有定义针对心跳的事件，可以借用其它事件，但消息中所带事件的 RequestID 一定要设为 0，以保证 MGC 能正确识别 MG 发的心跳消息，MGC 收到 MG 发的心跳消息时必须回正常响应。

建议在实现心跳机制时，应采用 MGC 和 MG 分别独立控制的心跳消息。

9.6 MGC-MG之间控制连接中断业务处理建议

依靠心跳机制，MGC 和 MG 都可以及时检测到对方中断。

9.6.1 MGC检测到中断

当 MGC 检测到 MG 中断后，将受控端点状态置为“ 断开 ”。从业务实现上应释放该 MG 上相关的呼叫状态和资源，停止计费。

9.6.2 MG检测到中断

当 MG 检测到 MGC 中断，将受影响端点置为“ 断开 ”，然后开始断开处理程序，为减少网络消息量，方法为“ disconnected ”的 RSIP 消息应采用通配符方式的端点标识。具体过程参见 9.4.7。

从业务上，当 MG 确认 MGC 中断后，可以选择将已有呼叫保持至与软交换连接恢复后收到软交换指令进行资源复位，也可以选择在呼叫保持一段时间后主动释放已建立的所有呼叫的资源，考虑到软交换还没来得及检测到与网关中断网络就已恢复的情况，呼叫保持的时间应不短于最大重传定时器值（2*T-HIST）与 MGC 心跳周期的和。

9.7 MGC-MG之间控制连接中断又恢复的处理建议

9.7.1 MGC检测到恢复

如果 MGC 在检测到中断后（9.6.1）收到了方法为“ disconnected ”的 RSIP 消息，或者是收到了对其心跳检测消息的响应，MGC 则断定控制连接恢复。此时，MGC 应立即采取措施和 MG 同步：在响应 MG 的 RSIP 消息的同时，发送 DLCX 命令释放中断前 MG 中存在的所有连接及相关端点。出于减少消息量的目的，在响应 MG 的 RSIP 消息的同时，可以只发送一条端点标识符为 all 的 DLCX 命令释放 MG 上所有连接及相关端点，但 MGC 从断定控制连接恢复起到收到该 DLCX 命令的正常响应之前，不应处理来自该网关的任何呼叫请求。

MGC 在等待 MG 对上述 DLCX 命令的响应时，如果收到了 MG 回 405 错误（The transaction could not be executed, because the endpoint is "restarting" ），则再次发送上述 DLCX 命令，直至 MG 回正常响应。

9.7.2 MG检测到恢复

当 MG 在进行断开处理程序时收到了 MGC 响应，MG 则断定控制连接恢复。此时 MG 如果收到上述 DLCX 命令应立即执行并回送响应。MG 对于端点标识符为 all 的 DLCX 命令应以通配符形式回送一个不带统计信息的正常响应，即使所涉及连接及端点早已经释放，也不需报错。

如果 MG 尚未收到 RSIP 消息的响应就收到了上述 DLCX 命令，则回 405 错误。

9.8 超长通话的审计

当呼叫进入通话阶段后，可能会发生网关中呼叫已释放，但 MGC 仍不知道的情况，譬如，网关在检测到网络中断后自行释放了呼叫，而 MGC 尚未来得及检测到网络中断，网络就已恢复。为了减轻这种情况对服务质量的影响，MGC 应对超长通话进行审计。

软交换事先设置超长通话保护时限，如半小时，当通话超过时限，MGC 对 MG 发送针对该端点及其连接的 AUCX 命令，如果 MG 中该连接仍存在，则回正常 200 响应表明通话仍在进行；如果 MG 中该连接已经删除，则回 515（The transaction refers to an incorrect connection-id (may have been already deleted)）错误码，MGC 收到该错误码就断定 MG 中的连接已删除，于是清除 MGC 中相关的呼叫状态和资源，停止计费。如果该 MG 对应呼叫的另一端在其它 MGC 的控制下，则 MGC 还应该向其它 MGC 发送对应的呼叫释放消息。

10 安全要求

任何实体都可以向一个 MGCP 端点发送命令。如果未授权的实体可以用 MGCP，他们就可以建立未经授权的呼叫，或者干涉合法呼叫。我们希望 MGCP 消息都应该在安全的 Internet 网络上传输。这种安全的 IP 网络架构正如 RFC2401 中定义的 IPsec，使用 RFC2402 定义的认证头（AH）或 RFC2406 定义的封装安全净荷（ESP）方式对协议的传输进行保护。

完整的 MGCP 协议栈结构如下：

如果网关和呼叫代理只接收由 Ipsec 提供认证的消息，就可以保护它们之间的连接。加密服务将提供对防止偷听的额外保护，这样可以防止已建立的合法连接受到监视。

如果像 SDP 中描述的那样，会话描述（session descriptions）被用于携带会话密钥，那么就需要加密服务。加密服务过程没有必要采取措施防止由非法网关和呼叫代理发起的拒绝服务攻击。

然而，他们将标识出这些非法实体，通过维护程序，这些非法实体将被取消授权。

10.1媒体连接的保护

本协议支持呼叫代理向网关提供会话密钥，用来对音频消息进行加密以防止窃听。

分组网络的一个缺陷就是容易受到由非法实体发起的恶意攻击。例如，非法实体可以通过向一个指定的 IP 地址和 UDP 端口发送大量 IP 包进行攻击。针对这种攻击进行保护的一种方法是仅接受来自获得授权实体的分组，例如，仅接受 Remote 描述符中列出的源 IP 地址和 UDP 端口号。然而这种方法可能会受到地址欺骗的影响以及导致连接建立的延迟。另外一种进行保护的方法是在呼叫建立过程中传送密钥对分组进行加密和认证，采用这种方法不会导致连接建立的延迟以及可以有效地防止地址欺骗。

11 包

如前所述，对本协议的扩展主要是通过包来实现的。本章介绍与包定义相关的一些要求。

一个包必定有唯一的一个名字。除了那些以“ x- ”或“ x+ ”开头的用于实验的包，所有包名必须经 IANA 注册才算有效。

每个包都一定有一个非负整数表示的版本，最初缺省的版本号是 0，每个新版本的版本号都在原有基础上加 1。新的版本必须向下兼容，即在新版的包里不许对老版的包做添加或删除，如果必须对包的内容进行添加或删除，就必须重新定义一个不同名的包。

带有类型为 timer-out 的信号的包可以标明是否包里所有的 time-out 信号和与这些信号都支持“ to ”参数及相关的缺省四舍五入规则，如果没有该参数的定义，就表明每个 time-out 信号都应该支持该定义。

一个包可以包括以下的一个或多个扩展选项。

* Actions
* BearerInformation
* ConnectionModes
* ConnectionParameters
* DigitMapLetters
* Events and Signals
* ExtensionParameters
* LocalConnectionOptions
* ReasonCodes
* RestartMethods
* Return codes

包的定义中对于每种所支持的扩展选项，都必须给出描述。注意，包的扩展选项与其它的扩展选项一样都必须遵循 MGCP 协议的语法规则。

11.1动作（Actions）

包所定义的扩展动作应该包括：

- 扩展动作名及其编码。
- 如果是带参数的扩展动作，就应该包括参数名及编码还有参数值。
- 扩展动作的行为描述。
- 可以与该扩展动作结合使用的协议中其它动作的列表。如果不提供该列表，就表明此扩展动作不能与协议中其它动作结合使用。
- 当包定义了多个扩展动作时，可以与该扩展动作结合使用的包中其它动作的列表。如果不提供该列表，就表明此扩展动作不能与包中其它动作结合使用。

不在一个包中定义的扩展动作最好不要同时被使用，除非已经对它们之间可能的相互干扰和负面影响考虑的非常周密。

11.2 承载信息（BearerInformation）

包所定义的扩展承载信息应该包括：

- 扩展承载信息名及其编码。
- 扩展承载信息可能的取值及其编码。
- 扩展承载信息的作用描述。如果要在 EndpointConfiguration 命令中省略扩展参数，那么就必须在该扩展承载信息作用描述里定义一个承载信息的缺省值。如果没有定义缺省值，那么在 EndpointConfiguration 命令中的 extension 参数将被置为空。

注意，扩展承载信息将被包含在 AuditEndpoint 命令审计 BearerInformation 的返回结果里。

11.3连接模式 （ConnectionModes）

包所定义的扩展连接模式应该包括：

- 扩展连接模式名及其编码。
- 扩展连接模式的作用描述。
- 采用扩展连接模式的连接与采用协议中定义的其它连接模式的连接之间的相互作用的描述。如果不提供这个描述，那么该扩展连接模式就不允许与其它连接模式相互作用。

由于包将会被放在一个包列表中报告上来，在审计 Capabilities 的 AuditEndpoint 命令的响应中的模式列表中将不会包含扩展连接模式。

11.4连接参数 （ ConnectionParameters）

包所定义的扩展连接参数应该包括：

- 扩展连接参数名及其编码。
- 扩展连接参数可能的取值及其编码。
- 扩展连接参数的起源的描述。

注意，扩展连接参数必须被包含在审计 connection parameters 的 AuditConnection 命令的返回结果中。

11.5号码表字母 （DigitMapLetters）

包所定义的扩展 DigitMapLetters 应该包括：

- 扩展 DigitMapLetters 名及其编码。
- 扩展 DigitMapLetters 含义的解释。

注意，在数图中的扩展 DigitMapLetters 并不遵守包所定义的扩展选项的命名习惯。具体来讲，在扩展 DigitMapLetters 的命名中不再用包名和斜杠（“/”）作为前缀，因此命名空间就变得单调且狭窄，而且可能会引起冲突。

因此，如果扩展 DigitMapLetters 的编码已经在一个包中被使用了，那么该扩展 DigitMapLetters 就不会在另一个包里被定义。如果在两个包里都使用相同的数图字母扩展编码，而且都被同一个端点支持，那么使用该书图字母扩展的后果将难以预料。

注意，尽管在数图里扩展 DigitMapLetter 的扩展名里没有包名和斜杠(“/”)，但是在满足 DigitMapLetter 的事件的事件代码作为观察到的事件被报告时，是包括包名前缀和斜杠的。换句话说，数图只是定义了数字采集匹配标准，但事件还是像其他事件一样被报告。

11.6事件和信号（ Events and Signals）

事件/信号的定义应该包括事件/信号的准确名称（即 MGCP 协议中使用的编码），对事件/信号的明文定义以及适当的时候给相应的事件/信号的精确定义，例如拨号音或 DTMF 音等话音信号的确切的频率。

对于每个事件/信号，包的定义必须包括以下信息：

- 事件/信号的定义和用途，其中应该包括用户实际产生的信号（例如，几十毫秒的 FSK 音）以及由此导致的用户观察到的结果（例如消息等待灯是开或是关）
- 事件/信号所使用的事件代码。
- 事件/信号的详细特性，如话音信号的频率和振幅，调制和重复。这些细节可能在国家规范里有规定。
- 可用事件/信号的典型持续时间和最大持续时间。
- 如果事件或信号可以（跨越媒体流）应用于一个连接，那么必须明确地指出。否则，就认为事件或信号不可以用于一个连接。
- 对于事件，必须提供以下信息：
- 表明事件是否是永久事件的标识。事件缺省地都不是永久性的，也不建议定义永久事件。注意永久事件的发现将自动触发一个 Notify 命令，除非呼叫代理明确规定端点采用不同的方式处理。这与 MGCP 通常的模式并不冲突，但仍然要假设呼叫代理支持现在正讨论的包。
- 表明事件是否该事件有可以统计的事件状态的标识。缺省地，事件没有可以统计的事件状态。

- 如果支持事件参数，就必须说明。而且必须（按照附录 A 中的语法）提供事件参数的准确的语法和语义。而且还要说明，是否可以把这些参数应用于 RequestedEvents, ObservedEvents, DetectEvents 和 EventStates 中，但是如果没有说明，则本协议假设：
- 这些参数不可以被应用于 RequestedEvents。
- 这些参数可以被应用于 ObservedEvents。
- 这些参数采用与 RequestedEvents 相同的方式被应用于 DetectEvents。
- 这些参数采用与 ObservedEvents 相同的方式被应用于 EventStates。
- 如果要在数图匹配中使用一个事件，就应该明确说明。注意只有使用单个的数字或字母做参数编码的事件才可以这样。

对于信号，必须提供以下信息：

- 信号类型（(OO, TO, BR)。
- Time-Out 信号应该值缺省的超时时间取值。有些情况下，超时时间值可能会改变。
- 如果支持信号参数，就必须明确指出。而且必须（按照附录 A 中的语法）提供信号参数的准确的语法和语义。
- Time-Out 信号可以指明是否支持“ to ”参数以及与 Time-Out 信号相关的四舍五入规则是什么。如果信号定义没有指明，就假设进行了“ 全包 ”的定义（package-wide definition）。如果包定义中没有定义是否支持“ to ”参数以及四舍五入规则，就将四舍五入规则缺省地置为与零最接近的时间非零秒数，而对于版本为 0 的包，“ to ”参数值缺省是“ no ” 而对于 0 以上版本的包，“ to ”参数缺省值为“ yes ”

根据以上描述，本协议给出推荐使用的事件和信号的定义形式：

Symbol	Definition	R	S Duration

其中，

- Symbol 表示事件/信号使用的事件代号，如“ hd ”。
- Definition 给出了事件/信号的简单描述。
- R 如果该事件是可以被发现的，R 就包含一个“ x ”，否则 R 就包含以下的一个或多个符号：

– “ P ” 表示事件是一个永久事件
– “ S ” 表示事件有一个可以被审计的事件状态。
– “ C ” 表示事件可以在一个连接上被发现。

- S 表示以下的信号类型：

– “ OO ” 表示信号是 On/Off 类型 。
– “ TO ”表示信号是 Time-Out 类型。

– “BR”表示信号是 Brief 类型。

- 另外 S 还包括：

– “C” 表识该信号可以被用于一个连接。

11.6.1 缺省和预留的事件

所有包含 Time-Out 类型的信号的包里都包含操作完成（operation complete （“oc”））和操作失败（operation failure （“of”））事件，而与包的描述中是否提供这两个事件无关。当要支持 Time-Out 类型的信号时这两个事件在支持是需要的，而且他们不能被 Time-Out 类型的信号覆盖。这些事件在需要时是可以被扩展的，但并不提倡这样做。

如果在没有 Time-Out 类型的信号的包里有“oc” 和“of”事件，那么包里提供的事件定义将覆盖这里所指的事件定义。这样的操作是不提倡的，而只是在避免后向兼容性问题时才会使用。

建议在缺省的事件定义中，明确地指出支持这两个事件，如下图：

Symbol	Definition	R	S Duration
oc	Operation Complete	x	
of	Operation Failure	x	

操作完成（oc）：当要求网关向端点或连接播放 TO 类型的信号时，这些 TO 类型的信号被成功的播放而没有因为收到了请求的事件（例如摘机或以拨数字事件）而被停止，只有这时才会生成操作完成事件。在报告操作完成事件的同时还应该携带播放完毕的信号名，例如，

O: G/oc(G/rt)

在以上的例子中，观察到的事件 oc 是在“G”包的“rt”信号播放完毕时引发的。

如果被报告的信号是用在一个连接上的，那么还应该携带该连接标识（Connection Id），例如，O: G/oc(G/rt@0A3F58)

当要求操作完成事件时，是无法确定任何事件参数的。当信号名里不包括包名时，就会将缺省的包认为是与事件相关的包。

操作失败(of)：当要求端点向端点或连接播放 TO 类型的信号时，这些 TO 类型的信号中的一个或多个信号在播放完毕前就出错了，这时才会生成操作失败事件。在报告操作完成事件的同时还应该携带出错的信号名，例如，

O: G/of(G/rt)

在上面的例子中，在处理“G”中的“rt”信号的过程中出错了。

如果被报告的信号是用在一个连接上的，那么还应该携带该连接标识（Connection Id），例如，O: G/of(G/rt@0A3F58)

当要求操作失败事件时，是无法确定任何事件参数的。当信号名里不包括包名时，就会将缺省的包认为是与事件相关的包。

11.7扩展参数（ExtensionParameters）

包所定义的扩展的 ExtensionParameters 应该包括：

- ExtensionParameters 的名及其编码。
- ExtensionParameters 可能的取值及其编码。
- 能说明对于协议定义的每条命令，ExtensionParameters 在请求和响应中是可选的，必选的还是禁止的的信息。注意，通常 ExtensionParameters 不是必选的。
- ExtensionParameters 的作用描述。作用描述中必须包括缺省值的定义，这样才可以在命令中省略 ExtensionParameters。如果作用描述中没有缺省值的定义，那么所有命令中的 ExtensionParameters 都被缺省的置为空。

是否该扩展选项可以通过 AuditEndpoint 和/或 AuditConnection 命令审计以及所返回的值。如果没有说明，那么只可以通过 AuditEndpoint 命令对扩展参数进行审计，而且返回是应该就是当前扩展选项的值，当然有可能是空值。

11.8本地连接选项（ LocalConnectionOptions）

包所定义的扩展的 LocalConnectionOptions 应该包括：

- 扩展本地连接选项名及其编码。
- 扩展本地连接选项可能的取值及其编码。
- 扩展本地连接选项的作用描述，其中必须包括以下方面的定义：

– 在 CreateConnection 命令里省略该扩展选项时，必须定义的缺省值。
– 在 ModifyConnection 命令里省略该扩展选项时，使用的缺省值。该值可能就是前面命令里的取值或者是上一条中定义的缺省值。如果没有说明，就会尽可能的保留当前值。
– 在审计 capabilities 时，是否会导致该扩展选项的返回和造成的影响的描述以及可能的返回值及其编码。如果这些 都没有说明，那么在审计 capabilities 时，就不应该返回该扩展选项。

另外，在通过 AuditConnection 命令审计 LocalConnectionOptions 时，必须返回该扩展选项。

11.9原因代码（ Reason Codes）

包所定义的扩展的原因代码应该包括：

- 扩展原因代码的数值。必须在 800 到 899 之间取值。
- 扩展原因代码的描述以及导致返回该原因代码的情况的描述。这种造成返回该原因值的情况应该限制在由这个包所定义的其它扩展选项所引起的事件中，这样才能确保事件的接收者能够正确地解析，得到原因值。

注意，在通过 AuditEndpoint 命令审计原因代码（reason code）时的返回结果中会提供扩展的原因代码。

11.10 重启方式（ RestartMethods）

包所定义的扩展的重启方式应该包括：

- 扩展的重启方式名及其编码。
- 该重启方式的描述，包括导致产生该重启方式的环境细节的描述。这些环境细节的应该被限定在包中定义的其它扩展选项所引起的事件上，这样事件的接收者才能解析并得到正确的扩展重启方式。
- 表明是否 RestartDelay 参数可以与该扩展重启方式一起使用的信息。如果没有说明，就假设它们无法一起使用，如果有 RestartDelay 参数，那它也会被忽略。
- 如果重启方式定义了一个服务状态，那么在重启方式的描述中一定要描述清除。这种情况下，在通过 AuditEndpoint 命令设计重启方式的结果中就会返回扩展的重启方式。

11.11 返回代码

包中定义的扩展的返回代码应该包括：

- 扩展的返回值的数值，必须在 800 到 899 之间。
- 扩展返回值的描述，包括造成返回该代码的情况描述。这些情况应该限定在由包里定义的其它扩展选项所引起的事件中，这样就可以保证接收事件者可以正确的解出扩展的返回代码。

12 流程

现网使用的相关设备必须支持本标准所示流程，但不限制设备制造商在此基础进行符合本标准和国际相关标准的业务扩充。

12.1注册流程

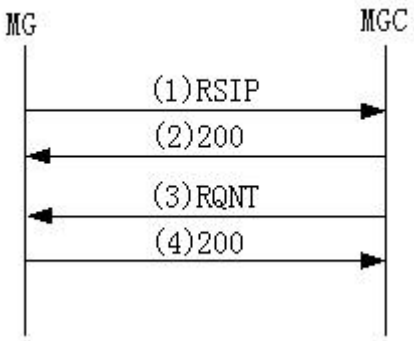


图 12-1：MG 注册流程图

流程说明：

1) MG 向 MGC 发 RSIP 命令，Endpoint 为*、mg 或具体的终结点，启动方式为 restart；

2) MGC 回响应；

3） MGC 下发检测摘机 RQNT 命令；请求 MG 检测摘机（L/HD）事件；

4） MG 回响应。

注：3)和 4)步只对 AG、IAD 有效。

如果是网关注册，Endpoint 应为* 或 mg，响应也应该只有一个，以减少网络消息负荷防止雪崩。

12.2注销流程

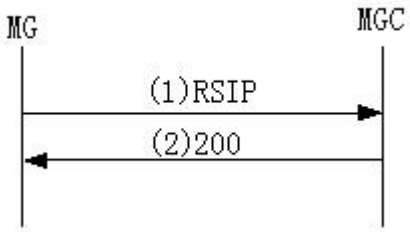


图 12-2：MG 注销流程图

流程说明：

1） MG 向 MGC 发 RSIP 命令，Endpoint 为*、mg 或具体的终结点，方式为 Forced；

2） MGC 回响应；

注：如果是网关注销，Endpoint 应为* 或 mg，响应也应该只有一个，以减少网络消息负荷防止雪崩。

12.3呼叫建立流程

12.3.1 AG-AG呼叫建立

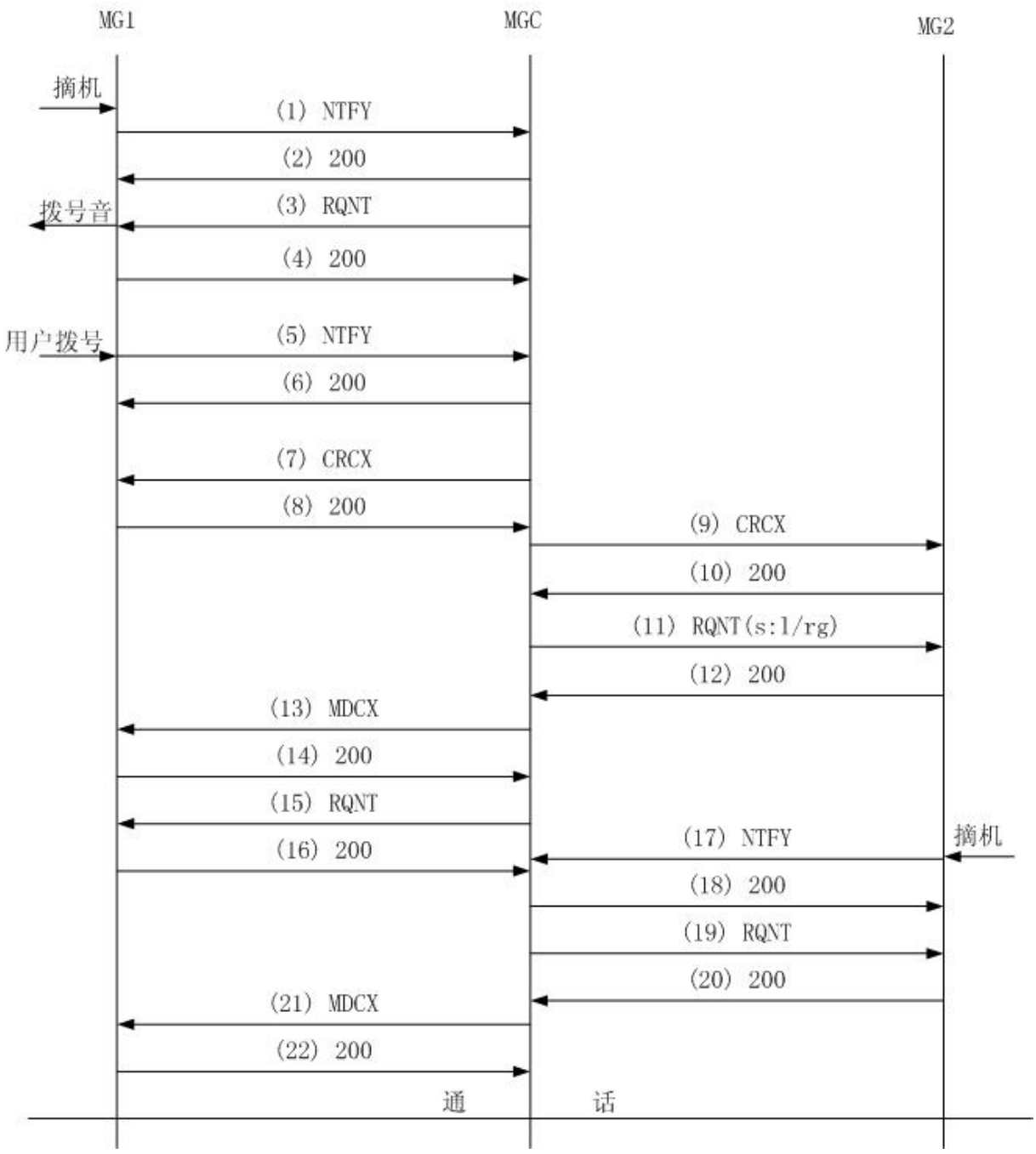


图 12-3：AG－AG 呼叫建立流程图

流程说明：

1） MG1 上 User1 摘机，MG1 发送 NTFY（L/HD）命令，通知 MGC；

2） MGC 回响应；

3） MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

4） MG1 回响应；

5） MG1 发送 NTFY 命令，将用户拨号送给 MGC；

6） MGC 回响应；

7） MGC 向 MG1 发送 CRCX 命令，为主叫创建一个连接，连接模式为 recvonly；

8） MG1 回响应，并将连接的 SDP 信息返回给 MGC；

9） MGC 向 MG2 发送 CRCX 命令，连接模式为 sendrecv，并且将主叫连接的 SDP 信息带给 MG2；

10）MG2 回响应，并将连接的 SDP 信息返回给 MGC；

11）MGC 向 MG2 发送 RQNT 命令，让被叫用户振铃（L/RG）；

12）MG2 回响应；

13）MGC 向 MG1 发送 MDCX 命令，把被叫的 SDP 信息带给 MG1；

14）MG1 回响应；

15）MGC 向 MG1 发送 RQNT 命令，主叫用户听回铃音；

16）MG1 回响应；

17）被叫用户摘机，MG2 发送 NTFY 命令给 MGC；

18）MGC 回响应；

19）MGC 向 MG2 发送 RQNT 命令，请求 MG2 监测挂机(L/HU)及拍叉簧(L/HF)；

20）MG2 回响应；

21）MGC 向 MG1 发送 MDCX 命令，修改连接模式为 sendrecv，并停回铃音；

22）MG1 回响应；主被叫通话；

注：如果定义摘/挂机等事件为永久事件，按本流程做相应改动。

12.3.2 TG-TG呼叫建立

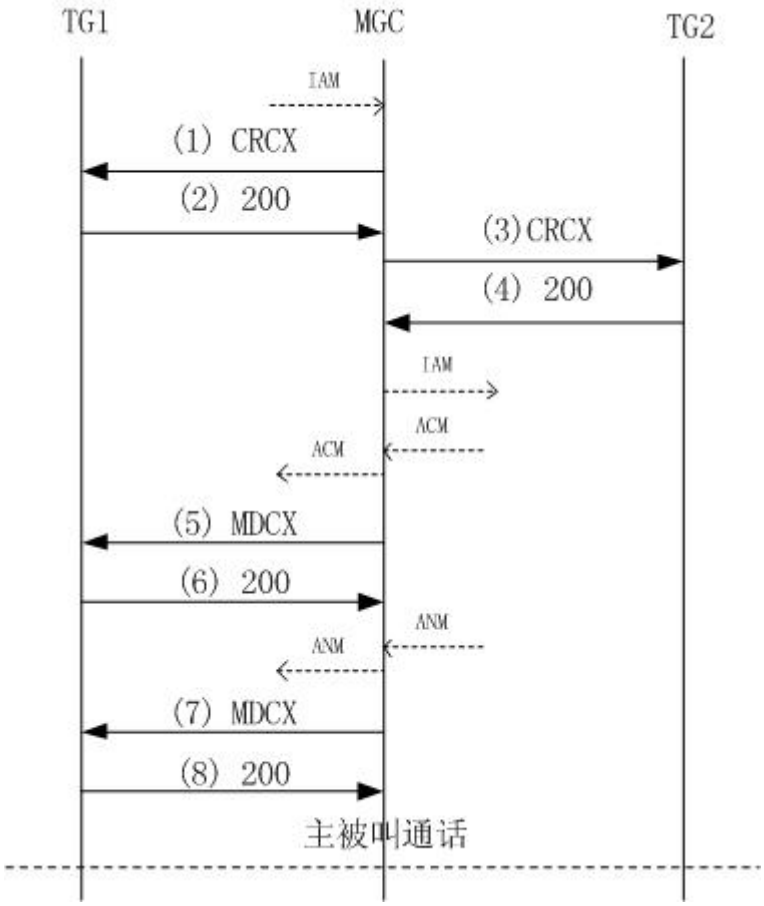


图 12-4：TG－TG 的呼叫建立流程图

流程说明：

1) 主叫用户摘机拨号后，MGC 收到主叫交换机的 ISUP 初始地址消息（IAM），然后向 TG1 发送 CRCX 消息，在 TG 中的入局中继上创建一个新连接，连接模式为 recvonly，并设置语音压缩算法；

2) TG1 向 MGC 回响应，其中包括 IP 地址，采用的语音压缩算法等；

3) MGC 向 TG2 发送 CRCX 消息，在 TG2 的出局中继上创建一个连接，连接模式为 sendrecv，并设置远端 RTP 地址及端口号、语音压缩算法等；

4) TG2 向 MGC 回响应，其中包括 IP 地址、端口号和采用的语音压缩算法等；MGC 收到 TG2 的正确响应后向被叫交换机发送 ISUP 初始地址消息（IAM）；

5) MGC 收到被叫交换机的 ISUP 地址全消息（ACM）后向 TG1 发送 MDCX 消息，设置远端 IP 地址、端口号和语音压缩算法等；

6) TG1 向 MGC 回响应。

7) MGC 收到被叫交换机的 ISUP 应答消息（ANM）后向 TG1 发送 MDCX 消息，将连接模式修改为 SendReceive；

8) TG1 向 MGC 回响应。

注：

1. 如果业务需要 TG 放回铃音，则需要在收到 ACM 消息后指示 TG 放回铃音，在收到 ANM 消息后，再指示 TG 停回铃音。
2. 出于节省消息考虑，如果不需要 TG 统计计费流量，可以接受在（5）同时将连接模式修改为 SendReceive，而省去（7）、（8）两条消息。

12.4呼叫释放流程

12.4.1 AG-AG呼叫释放

12.4.1.1 互不控释放

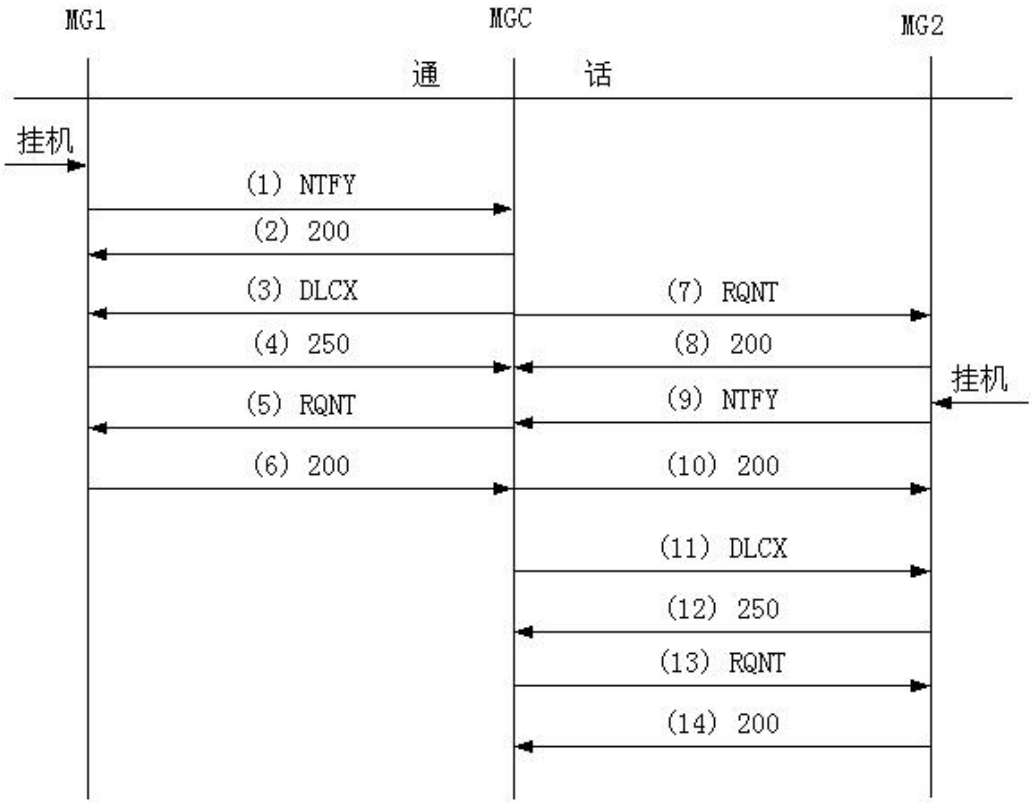


图 12-5：互不控呼叫释放流程图

流程说明：

1） MG1 挂机，向 MGC 发送 NTFY 命令；

2） MGC 回响应；

3） MGC 向 MG1 发送 DLCX 命令，拆除对应连接；

4） MG1 回响应，其中包括上报统计信息；

5） MGC 向 MG1 发送 RQNT 命令，监视摘机（L/HD）；

6） MG1 回响应；

7） MGC 向 MG2 发送 RQNT 命令，让 MG2 放忙音（L/BZ）；

8） MG2 回响应；

9） 被叫挂机，MG2 向 MGC 发送 NTFY 命令；

10）MGC 回响应；

11）MGC 向 MG2 发送 DLCX 命令，拆除对应连接；

12）MG2 回响应，其中包括上报统计信息；

13）MGC 向 MG2 发送 RQNT 命令，监视摘机（L/HD）；

14）MG2 回响应；

12.4.1.2 主叫控制释放

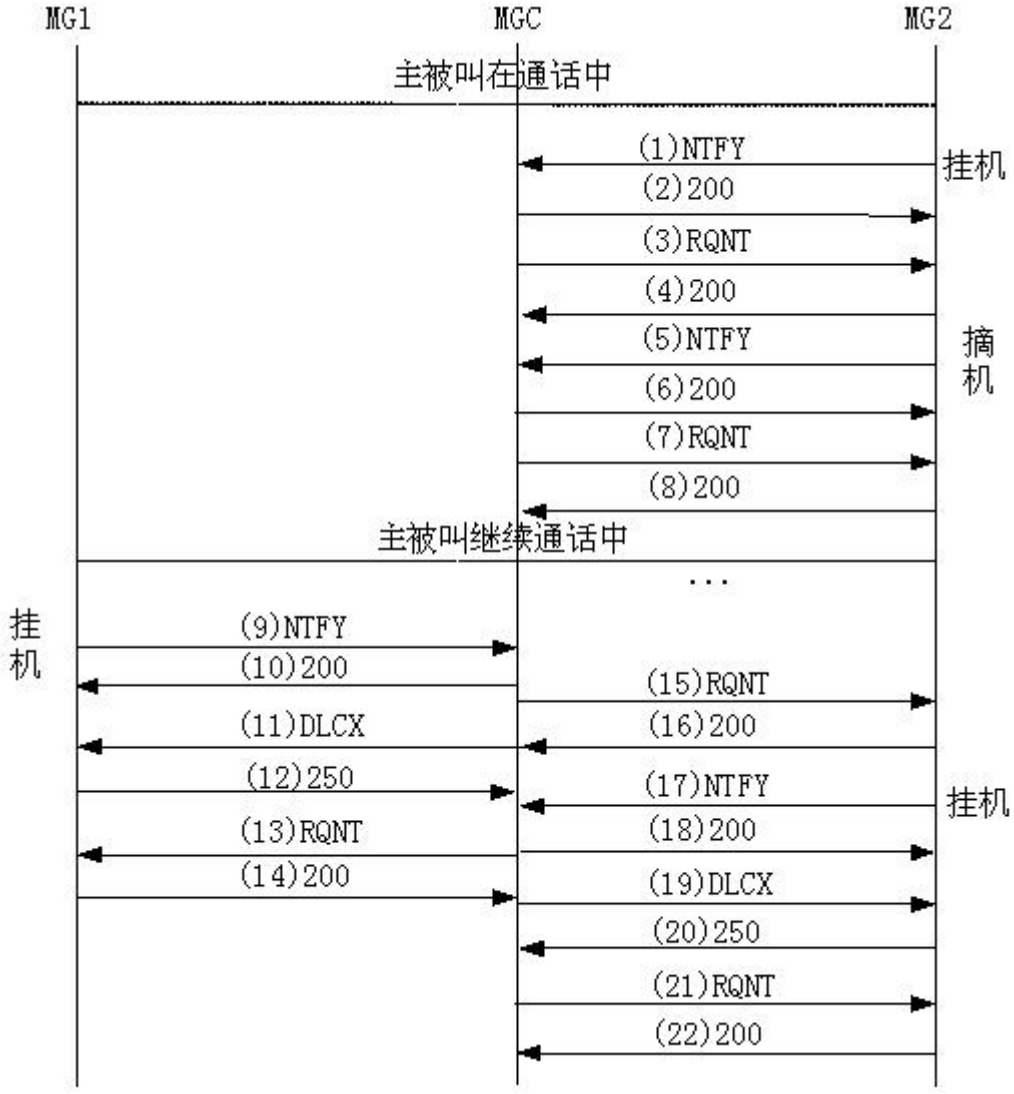


图 12-6：主叫控制释放流程图

流程说明：（主被叫进入语音通话之前的信令流程与普通拨号通话流程一致，MG1 为主叫）

1） MG2 挂机，MG2 向 MGC 发送 NTFY 命令；

2） MGC 回响应；

3） MGC 向 MG2 发送 RQNT，请求被叫 MG2 检测摘机(L/HD)；

4） MG2 回响应；

5） 被叫摘机，MG2 向 MGC 发送 NTFY 命令；

6） MGC 回响应；

7） MGC 向 MG2 发送 RQNT，请求 MG2 检测挂机(L/HU)、拍叉簧(L/HF)；

8） MG2 回响应，主被叫继续通话；

被叫可以多次重复(1)~~(8)流程。

9） 主叫挂机，MG1 向 MGC 发送 NTFY 命令；

10）MGC 回响应；

11）MGC 向 MG1 发送 DLCX 命令，拆除对应连接；

12）MG1 回响应，其中包括上报统计信息；

13）MGC 向 MG1 发送 RQNT，请求 MG1 检测摘机(L/HD)；

14）MG1 回响应；

15）MGC 向 MG2 发送 RQNT，请求 MG1 对被叫放忙音；

16）MG2 回响应；

17）被叫挂机，MG2 向 MGC 发送 NTFY 命令；

18）MGC 回响应；

19）MGC 向 MG2 发送 DLCX 命令，拆除对应连接；

20）MG2 回响应，其中包括上报统计信息；

21）MGC 向 MG2 发送 RQNT 命令，监视摘机 L/HD；

22）MG2 回响应；

12.4.1.3 被叫控制释放

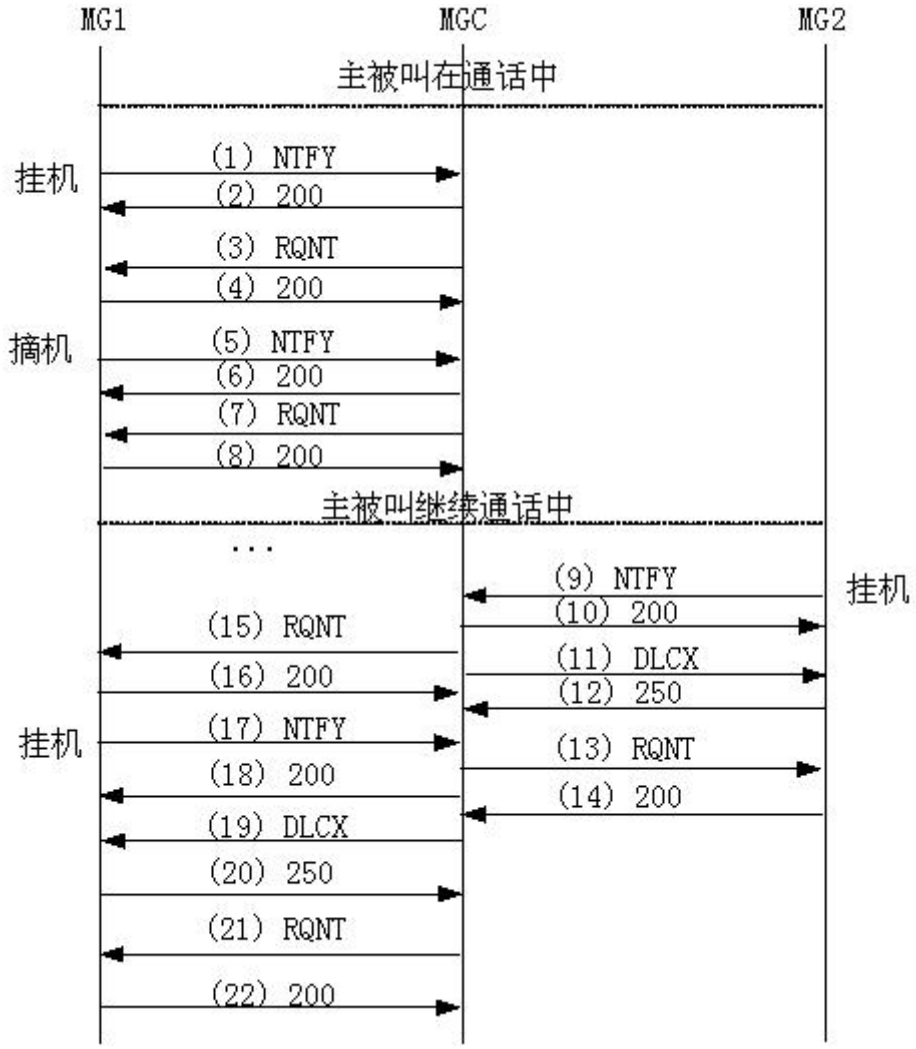


图 12-7：被叫控制释放流程图

流程说明：（主被叫进入语音通话之前的信令流程与普通拨号通话流程一致，MG2 为被叫）

1） MG1 挂机，MG1 向 MGC 发送 NTFY 命令；

2） MGC 回响应；

3） MGC 向 MG1 发送 RQNT，请求 MG1 检测摘机(L/HD)；

4） MG1 回响应；

5） 主叫摘机，MG1 向 MGC 发送 NTFY 命令；

6） MGC 回响应；

7） MGC 向 MG1 发送 RQNT，请求 MG1 检测挂机(L/HU)、拍叉簧(L/HF)；

8） MG1 回响应，主被叫继续通话；

主叫可以多次重复(1)~~(8)流程。

9） 被叫挂机，MG2 向 MGC 发送 NTFY 命令；

10）MGC 回响应；

11）MGC 向 MG2 发送 DLCX 命令，拆除对应连接；

12）MG2 回响应，其中包括上报统计信息；

13）MGC 向 MG2 发送 RQNT，请求 MG2 检测摘机(L/HD)；

14）MG2 回响应；

15）MGC 向 MG1 发送 RQNT，请求 MG1 对主叫放忙音；

16）MG1 回响应；

17）主叫挂机，MG1 向 MGC 发送 NTFY 命令；

18）MGC 回响应；

19）MGC 向 MG1 发送 DLCX 命令，拆除对应连接；

20）MG1 回响应，其中包括上报统计信息；

21）MGC 向 MG1 发送 RQNT 命令，监视摘机 L/HD(N)；

22）MG1 回响应；

12.4.2 TG-TG呼叫释放

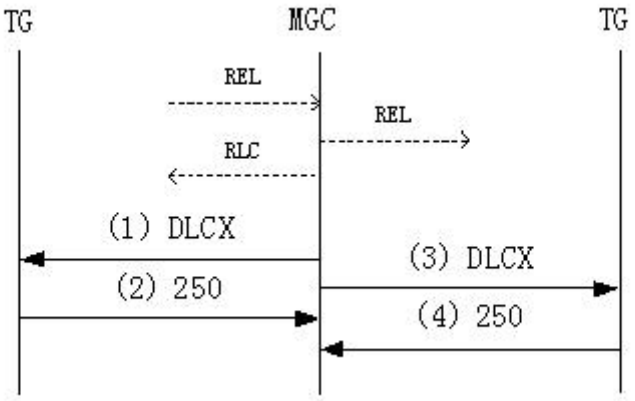


图 12-8：TG－TG 呼叫释放流程图

流程说明：

1） 主叫的挂机，主叫交换机向 MGC 发送 ISUP 释放消息（REL），并向 TG1 发送 DLCX 消息，删除连接；

2） TG1 向 MGC 回响应，并上报统计信息；

3） MGC 向被叫交换机发送 ISUP 释放消息（REL），并向主叫发送 ISUP 释放完成消息（RLC），MGC 向 TG2 发送 DLCX 消息，释放连接；

4） TG2 向 MGC 回响应，并上报统计信息。

12.5放通知音流程

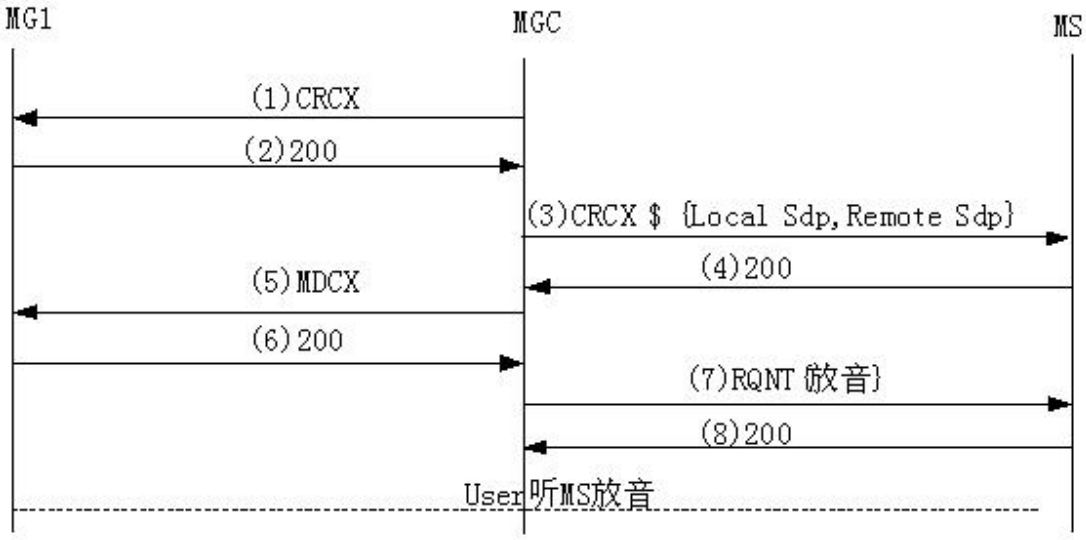


图 12-9：放通知音流程图

流程说明：（MGC 分析到需对 MG1 上的用户放音）

1） MGC 向 MG1 发送 CRCX 消息,在 MG1 中创建一个新的连接，连接的 Mode 设置为 ReceiveOnly；

2） MG1 向 MGC 返回响应；

3） MGC 分析 MG1 上 User1 的业务状态，发现需对其放音，向 MS 发送 CRCX 消息，在 MS 中创建一个新的连接，连接的端点可以设为$，也可以由 MGC 指定，设置连接的 Mode 设置为 SendRecv，远端 SDP 为 MG1 上 User1 对应 RTP 属性；

4） MS 向 MGC 返回响应；

5） MGC 向 MG1 发送 MDCX 消息，设置 MG1 对应的远端 SDP 属性为 MS 的 RTP 属性。

6） MG1 向 MGC 返回响应；

7） MGC 向 MS 发送 RQNT 消息，让其放音。

8） MS 向 MGC 返回响应，并对 User1 放音；

注：如果 MG1 上的 User1 已经建立了连接端口，则 1)、2）步可以省略

12.6 MGC-MG之间异常呼叫流程

12.6.1 久不拨号

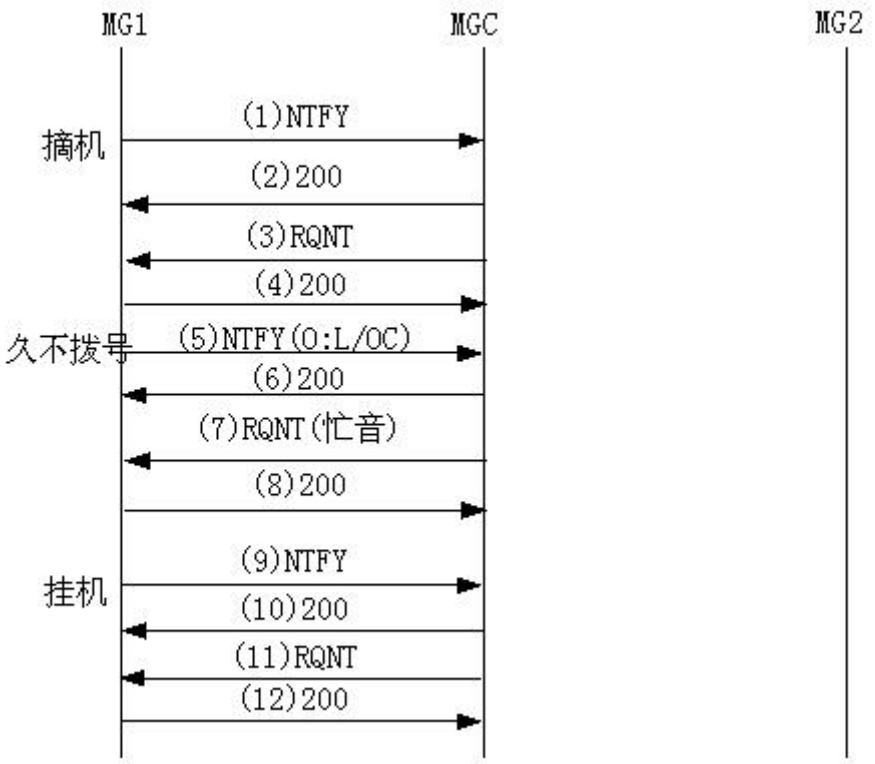


图 12-10：久不拨号流程图

流程说明：

1) 主叫摘机，MG1 发送 NTFY 命令，通知 MGC；

2) MGC 回响应；

3) MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求监测用户拨号、检测挂机 L/HU、拍叉簧 L/HF、放音结束事件 L/OC；

4) MG1 回响应；

5) 用户久不拨号超时，MG1 发送 NTFY 命令，将拨号音完成事件 L/OC 送给 MGC；

6) MGC 回响应；

MGC 分析主叫用户是否有延迟热线业务，如果有则取出热线被叫号码按正常呼叫流程继续；如果没有则按异常处理流程（如下），本节只详列异常处理流程：

7) MGC 向 MG1 发送 RQNT 事件，放忙音；

8) MG1 回响应；

9) 主叫挂机，MG1 向 MGC 发送 NTFY 命令

10) MGC 回响应

11) MGC 向 MG1 发送 RQNT 命令，监视摘机 L/HD；

12) MG1 回响应；

12.6.2 空号

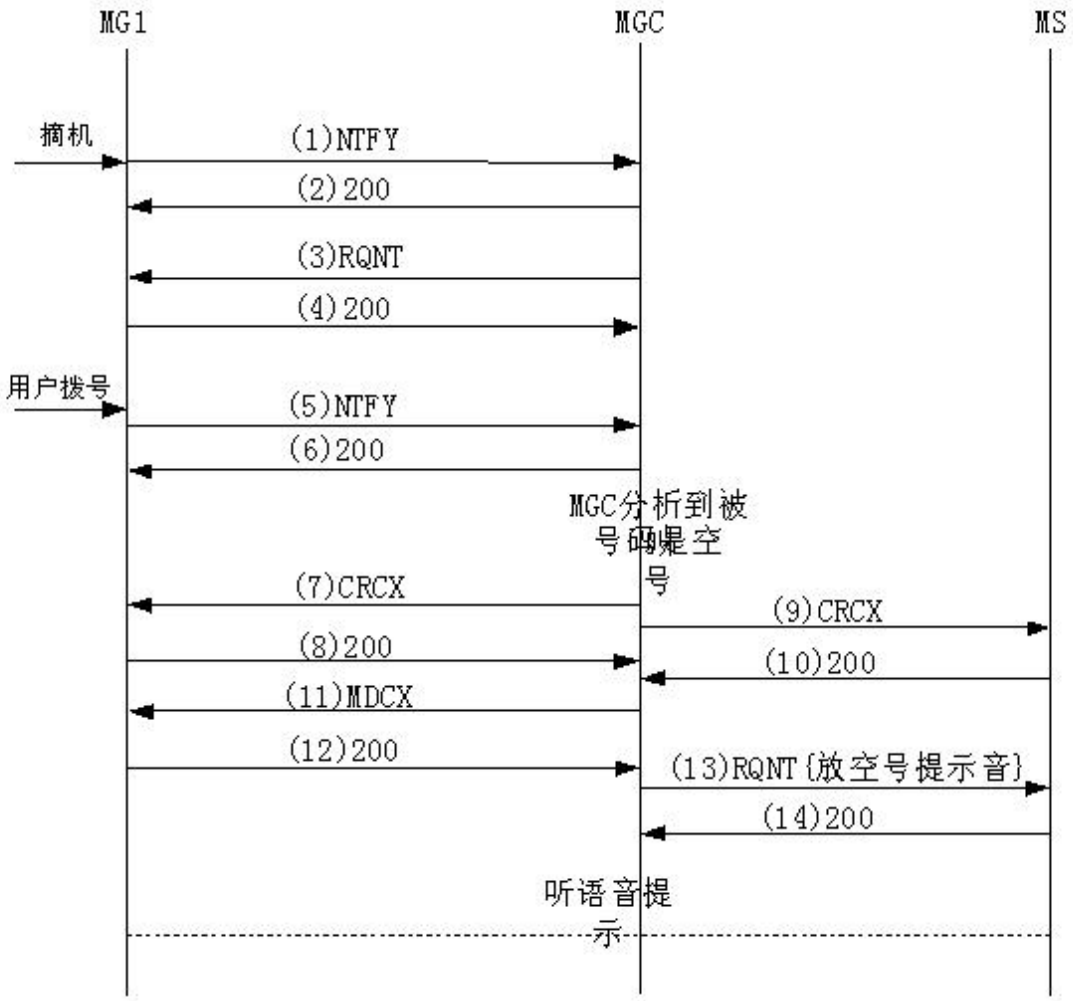


图 12-11：空号流程图

流程说明：

1) MG1 检测到用户 User1 的摘机消息，将此摘机事件通过 NTFY 命令上报给 MGC；

2) MGC 向 MG1 返回响应；

3) MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求监测用户拨号、检测挂机 L/HU、拍叉簧 L/HF、放音结束事件 L/OC；

4) MG1 向 MGC 返回响应；

5) MG1 上的用户 User1 拨号，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码用 NTFY 消息上报 MGC；

6) MGC 向 MG1 返回响应；

MGC 分析到被叫号码是空号。

7) MGC 向 MG 发送 CRCX 消息，在 MG 中为用户端点创建一个连接，设置连接的 Mode 设置为

ReceiveOnly；

8） MG1 向 MGC 返回响应，其中包括该 RTP1 的 IP 地址，采用的语音压缩算法和 RTP 端口号等。

9） MGC 向 MS 发送 CRCX 消息申请放音资源，端点为$或由 MGC 指定，申请创建一个连接；并设置连接的 Mode 为 SendRecv；设置语音压缩算法；并携带远端 SDP 为 RTP1。

10）MS 向 MGC 返回响应，携带为连接分配的 SDP 信息 Y；

11）MGC 向 MG 发送 MDCX 消息，携带远端 SDP 信息 Y；

12）MG 向 MGC 返回响应；

13）MGC 向 MS 下发 RQNT 消息，让 MS 放空号提示音；

14）MS 向 MGC 返回响应；

12.6.3 错号

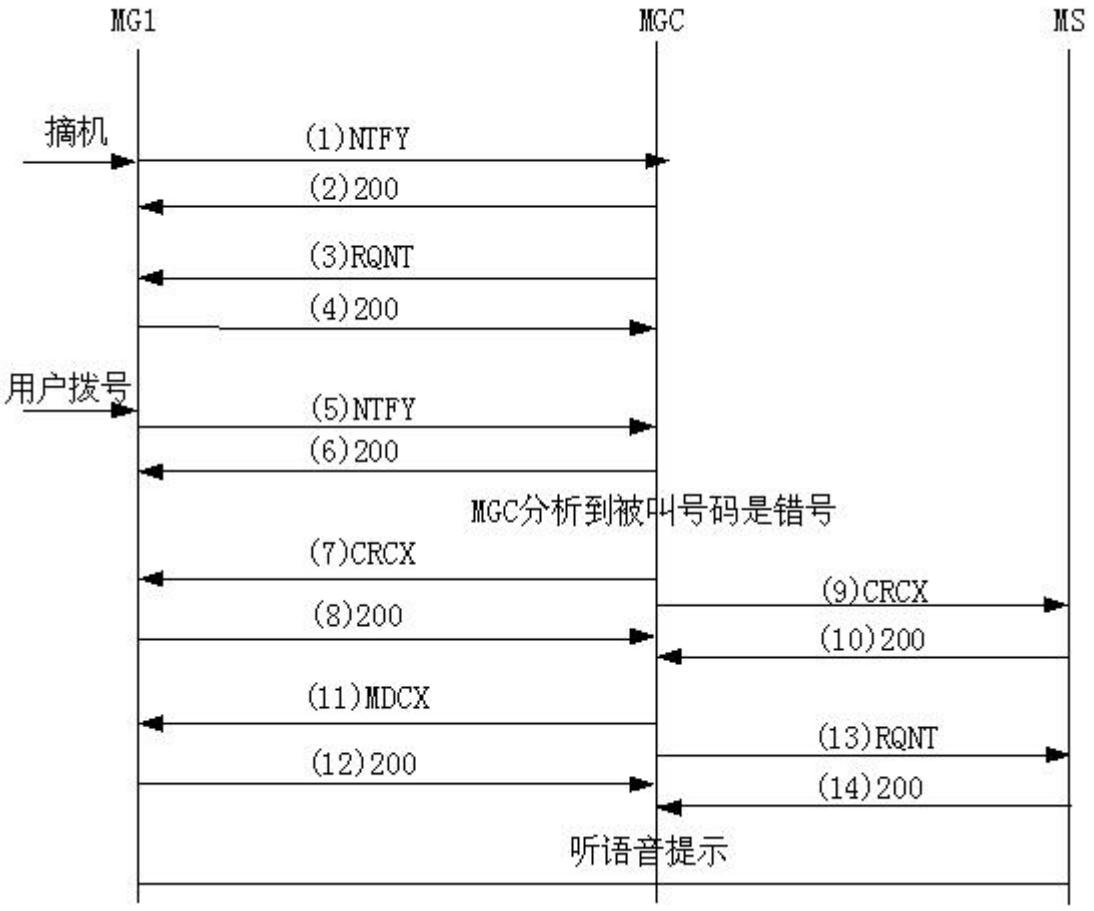


图 12-12：错号流程图

流程说明：

1） MG1 检测到用户 User1 的摘机消息，将此摘机事件通过 NTFY 命令上报给 MGC；

2） MGC 向 MG1 返回响应；

3） MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求监测用户拨号、检测挂机 L/HU、拍叉簧 L/HF、放音结束事件 L/OC；

4） MG1 向 MGC 返回响应；

5） MG1 上的用户 User1 拨号，MG1 根据 MGC 所下发的号码表进行收号：如果第一位号码匹配失败，则上报第一位号码；如果拨号中间匹配失败或超时，则上报用户失败前所拨的号码加发生匹配失败的那位号；如果拨号完全匹配，则上报所拨号码；

6） MGC 向 MG1 返回响应；

MGC 分析到用户拨错号（包括未拨全号）：

7） MGC 向 MG 发送 CRCX 消息，在 MG 中为 User1 创建一个连接，连接模式为 recvonly；

8） MG1 回响应，并将连接的 SDP 信息返回给 MGC；

9） 向 MS 发送 CRCX 消息，在 MS 中创建一个新的连接，连接的端点可以设为$，也可以由 MGC 指定，设置连接的 Mode 设置为 SendRecv，远端 SDP 为 MG1 上 User1 对应 RTP 属性；

10） MS 向 MGC 返回响应；

11） MGC 向 MG1 发送 MDCX 消息，设置 MG1 对应的远端 SDP 属性为 MS 的 RTP 属性。

12） MG1 向 MGC 返回响应；

13） MGC 向 MS 发送 RQNT 消息，让其放错号提示音；

14） MS 向 MGC 返回响应，并对 User1 放音。

12.6.4 后挂方久不挂机

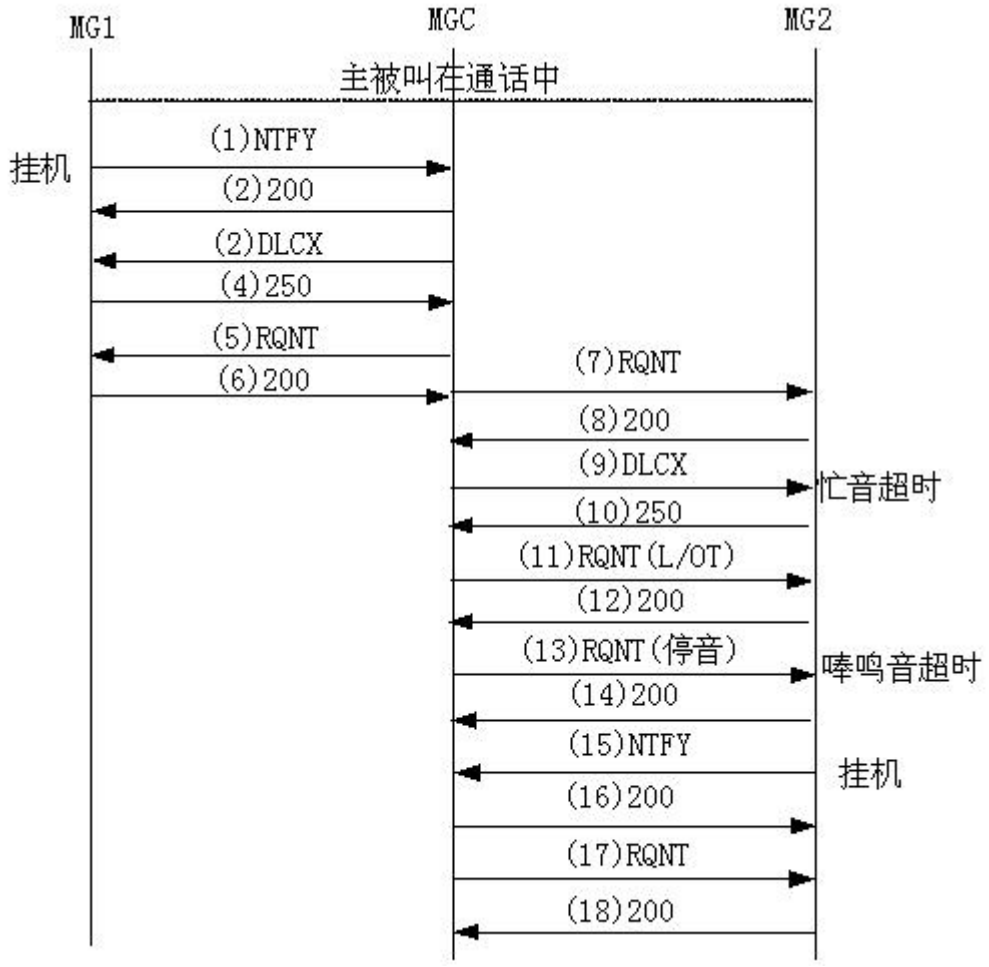


图 12-13：被叫久不挂机流程图

流程说明：（主被叫进入语音通话之前的信令流程与普通拨号通话流程一致，MG1 为主叫）

1） 主叫挂机，MG1 向 MGC 发送 NTFY 命令；

2） MGC 回响应；

3） MGC 向 MG1 发送 DLCX 命令，拆除对应连接；

4） MG1 回响应；

5） MGC 向 MG1 发送 RQNT 命令，监视摘机 L/HD；

6） MG1 回响应；

7） MGC 向 MG2 发送 RQNT 命令，送忙音；

8） MG2 回响应；

9） 放忙音超时，MGC 向 MG2 发送 DLCX 命令，拆除对应连接；

10）被叫 MG 回响应

11）MGC 向 MG2 发送 RQNT 命令，送嗲鸣音(L/OT)；

12）MG2 回响应；

13）嗏鸣音超时，MGC 向 MG2 发送 RQNT 命令，停嗏鸣音；

14）MG2 回响应；

15）被叫挂机，MG2 向 MGC 发送 NTFY 命令；

16）MGC 回响应；

17）MGC 向 MG2 发送 RQNT 命令，监视摘机 L/HD；

18）MG2 回响应；

12.7补充业务流程

12.7.1 呼叫前转

由于立即前转和遇忙前转对 MGCP 协议来说与普通呼叫流程无异，本文仅列无应答前转流程。

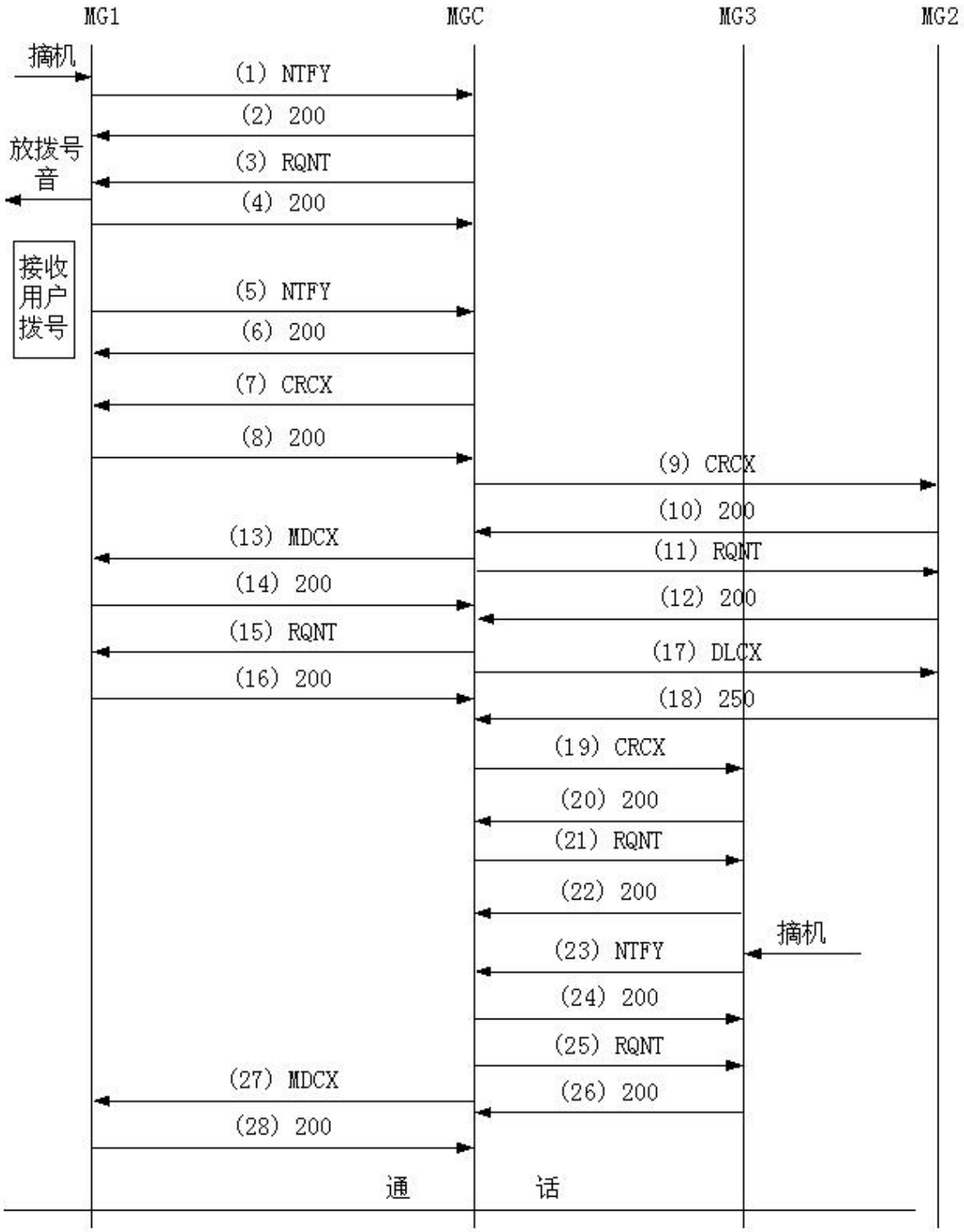


图 12-14： 无应答前转流程

流程说明：

1) 主叫摘机，MG1 发送 NTFY 命令，通知 MGC；

2) MGC 回响应；

3） MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

4） MG1 回响应；

5） MG1 发送 NTFY 命令，将用户拨号送给 MGC；

6） MGC 回响应；

7） MGC 向 MG1 发送 CRCX 命令，为主叫创建一个连接，连接模式为 recvonly；

8） MG1 回响应，并将连接的 SDP 信息返回给 MGC；

9） MGC 进行号码分析，向 MG2 发起呼叫。MGC 向 MG2 发送 CRCX 命令，连接模式为 sendrecv，并且将主叫连接的 SDP 信息带给被叫 MG2；

10） MG2 回响应，并将连接的 SDP 信息返回给 MGC；

11） MGC 向 MG2 发送 RQNT，让被叫振铃；

12） MG2 回响应；

13） MGC 向 MG1 发送 MDCX 命令，把 MG2 的 SDP 带给主叫；

14） MG1 回响应；

15） MGC 向 MG1 发送 RQNT 命令，主叫用户听回铃音；

16） MG1 回响应；

17） MG2 久不摘机，MGC 进行号码分析，发现 MG2 上的该用户设置了无应答转移到 MG3，MGC 向 MG2 发送 DLCX 命令；

18） MG2 回响应；

19） MGC 向 MG3 发送 CRCX 命令，连接模式为 sendrecv，并且将主叫连接的 SDP 信息带给 MG3；

20） MG3 回响应，并将连接的 SDP 信息返回给 MGC；

21） MGC 向 MG3 发送 RQNT，让被叫振铃；

22） MG3 回响应；

23） 被叫用户摘机，MG3 发送 NTFY 命令；

24） MGC 回响应；

25） MGC 向 MG3 发送 RQNT 命令，请求监测挂机及拍叉簧；

26） MG3 回响应；

27） MGC 向 MG1 发送 MDCX 命令，停回铃音，修改连接模式为 sendrecv，并将 MG3 的 SDP 信息带给 MG1；

28） MG1 回响应；

12.7.2 主叫号码显示

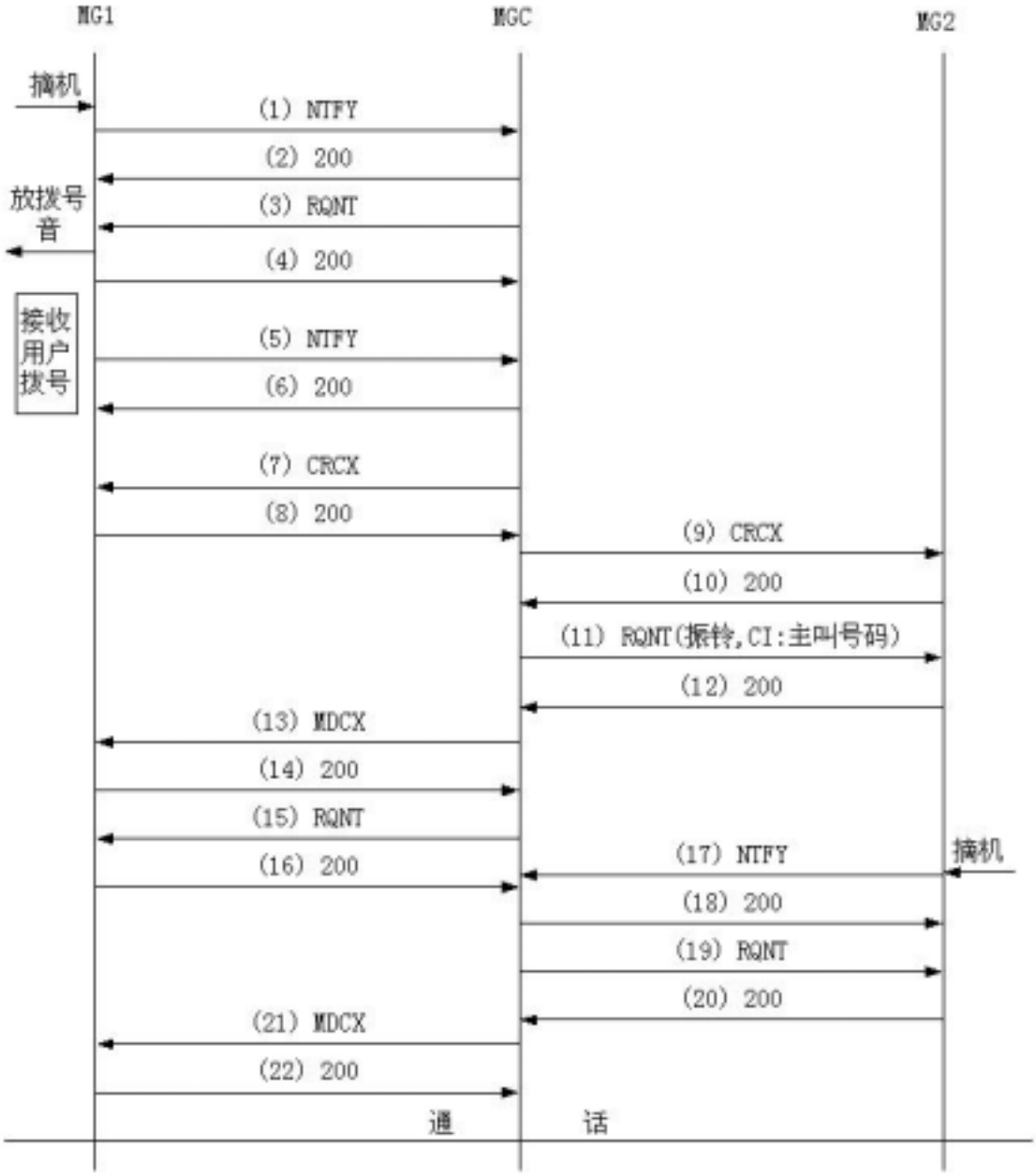


图 12-15：主叫号码显示流程图

流程说明：

1) MG1 上 User1 摘机，MG1 发送 NTFY（L/HD）命令，通知 MGC；

2) MGC 回响应；

3) MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求监测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

4) MG1 回响应；

5) MG1 发送 NTFY 命令，将用户拨号送给 MGC；

6） MGC 回响应；

7） MGC 向 MG1 发送 CRCX 命令，为主叫创建一个连接，连接模式为 reciveonly；

8） MG1 回响应，并将连接的 SDP 信息返回给 MGC；

9） MGC 向 MG2 发送 CRCX 命令，连接模式为 sendrecv，并且将主叫连接的 SDP 信息带给被叫 MG；

10）MG2 回响应，并将连接的 SDP 信息返回给 MGC；

11）MGC 向 MG2 发送 RQNT 命令，让被叫用户振铃，并通过 L/CI 把主叫号码带给 MG2；

12）MG2 回响应；

13）MGC 向 MG1 发送 MDCX 命令，把被叫的 SDP 信息带给主叫 MG；

14）MG1 回响应；

15）MGC 向 MG1 发送 RQNT 命令，主叫用户听回铃音；

16）MG1 回响应；

17）被叫用户摘机，MG2 发送 NTFY 命令给 MGC；

18）MGC 回响应；

19）MGC 向 MG2 发送 RQNT 命令，请求被叫 MG 监测挂机(L/HU)及拍叉簧(L/HF)；

20）MG2 回响应；

21）MGC 向 MG1 发送 MDCX 命令，修改连接模式为 sendrecv，并停回铃音；

22）MG1 回响应；主被叫通话；

12.7.3 呼叫等待

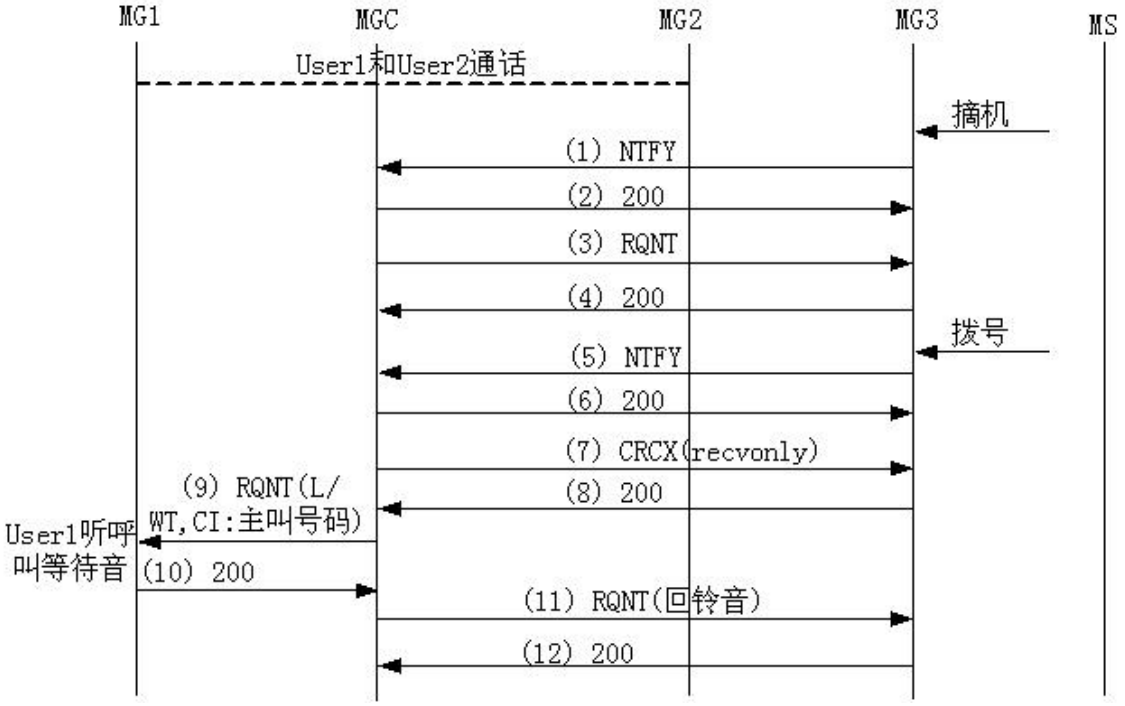


图 12-16：呼叫等待第三方呼入流程图

流程说明：

USER1（有呼叫等待业务）与 USER2 已经在通话中；USER3 摘机；

1) MG3 向 MGC 送 NTFY 命令，上报 USER3 摘机事件；

2) MGC 回送响应；

3) MGC 向 MG3 送 RQNT 消息,请求 MG3 放拨号音,下发拨号表并要求检测用户拨号、挂机(L/HU)、拍叉簧（L/HF）及放音结束事件（L/OC）；

4) MG3 回送响应；等待用户拨号，用户 USER3 听拨号音；

5) MG3 上的用户 User3 拨号，MG3 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

6) MGC 回送响应；

7) MGC 向 MG3 发送 CRCX 命令，为主叫创建一个连接，连接模式为 recvonly，并设置其语音压缩算法;

8) MG3 向 MGC 回送响应，并带回 SDP 信息，包括 IP 地址，采用的语音压缩算法和语音端口号等。

9) MGC 分析 USER1 有呼叫等待业务，向 MG1 发送 RQNT 命令，通知 MG1 向 USER1 送呼叫等待音（L/WT）、检测拍叉簧事件，如果 USER1 有主叫号码显示权限，还必须通过（L/CI）带给 MG1 主叫号码；

10) MG1 向 MGC 回送响应；用户 USER1 听呼叫等待音；

11) MGC 向 MG3 送 RQNT 命令，通知 MG3 向 USER3 送回铃音；

12）MG3 向 MGC 回送响应，USER3 听回铃音；

至此，USER1 可以选择继续与 USER2 通话，还是与新用户 USER3 通话。

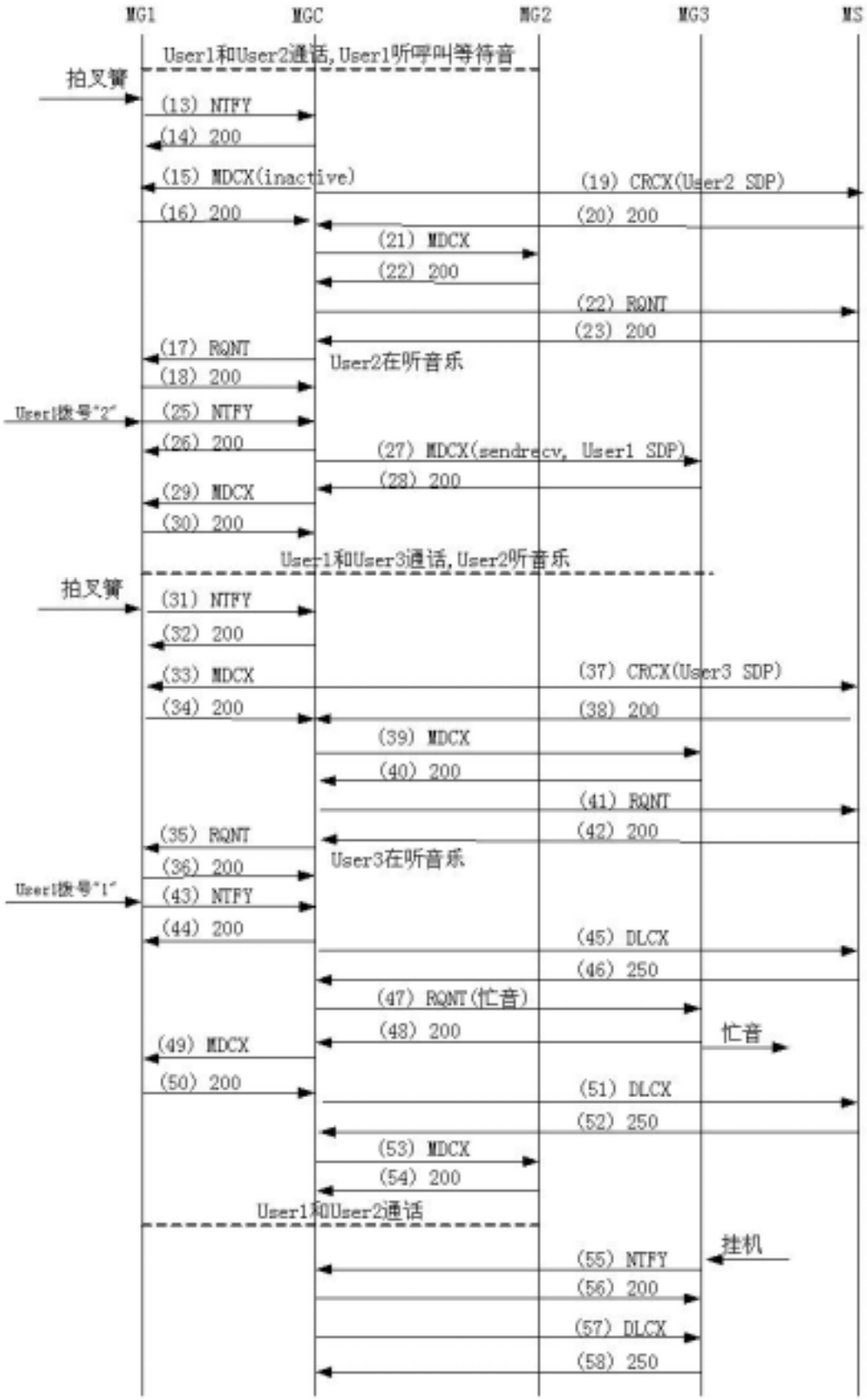


图 12-17：呼叫等待用户选择业务键流程图

13）MG1 检测到 USER1 拍叉簧事件，用 NTFY 上报给 MGC;

14）MGC 回送响应；

15）MGC 向 MG1 发送 MDCX 命令，将 USER1 的模式修改为 inactive；

16）MG1 回送响应；

17）MGC 向 MG1 送 RQNT 命令，向 MG1 发送号码表；请求 MG1 放拨号音；并检测收号完成、挂机事件；

18）MG1 向 MGC 回送响应；对用户 USER1 放拨号音，等待收号；

19）MGC 使用 CRCX 命令向 MS 申请资源，附带 USER2 的 SDP 信息,用来对 USER2 放音乐；

20）MS 向 MGC 回送响应，携带分配的资源终端 Y1 的 SDP 信息；

21）MGC 向 MG2 发送 MDCX 命令修改 USER2 的 SDP 信息，使其和 Y1 相通，同时修改其模式为 ReciveOnly；

22）MG2 回送响应；

23）MGC 向 MS 发送 RQNT 命令，要求 Y1 向外部发送音乐；

24）MS 向 MGC 回送响应。这时 User2 处于听音乐状态；

用户 USER1 拨号‘ 2 ’，同 USER3 通话，同时保留与 USER2 的呼叫。

25）MG1 上的用户 User1 拨号，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

26）MGC 回送响应；

27）MGC 向 MG3 送 MDCX 命令，通知 MG3 修改 USER3 的远端媒体属性(对应 USER1 媒体属性)，并将模式修改为修改为 SendReceive；

28）MG3 回送响应；

29）MGC 向 MG1 送 MDCX 命令，通知 MG1 修改 USER1 的远端媒体属性(对应 USER3 媒体属性)；

30）MG1 回送响应；

此时用户 USER1 同用户 USER3 通话，用户 2 听音乐；如果用户 USER1 想恢复同 USER2 的会话，截断同 USER3 的会话，继续拍叉簧。

31）用户 USER1 拍叉簧，MG1 检测到拍叉簧事件，用 NTFY 上报给 MGC；

32）MGC 回送响应；

33）MGC 向 MG1 发送 MDCX 命令，将 USER1 的模式修改为 inactive；

34）MG1 回送响应；

35）MGC 向 MG1 送 RQNT 命令，向 MG1 发送号码表；请求 MG1 放拨号音；并检测收号完成、挂机事件；

36）MG1 向 MGC 回送响应；对用户 USER1 放拨号音，等待收号；

37）MGC 使用 CRCX 命令向 MS 申请资源，附带 USER3 的 SDP 信息,用来对 USER3 放音乐；

38）MS 向 MGC 回送响应，携带分配的资源终端 Y2 的 SDP 信息；

39）MGC 向 MG3 发送 MDCX 命令修改 USER3 的 SDP 信息，使其和 Y2 相通，同时修改其模式为 ReciveOnly；

40）MG3 回送响应

41）MGC 向 MS 发送 RQNT 命令，要求 Y2 向外部发送音乐；

42）MS 向 MGC 回送响应。这时 User3 处于听音乐状态；

USER1 拨号‘1’，切断与当前用户 USER3 的通话，而回到与 USER2 的通话中。

43）用户 User1 拨号,MG1 根据 MGC 所下发的号码表进行收号,并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

44）MGC 回送响应；

45）MGC 向 MS 发送 DLCX 命令，要求释放 Y2,停止对 User3 放音；

46）MS 回送响应；

47）MGC 向 MG3 送 RQNT 命令，通知 MG3 对 USER3 放忙音；

48）MG3 回送响应；用户 USER3 听忙音；

49）MGC 向 MG1 送 MDCX 命令，通知 MG1 修改 USER1 的远端媒体信息为 USER2 的媒体属性,同时模式修改为 sendrecv。

50）MG1 回送响应；

51）MGC 向 MS 发送 DLCX 命令，释放 Y1,停止对 User2 放音；

52）MS 回送响应；

53）MGC 向 MG2 送 MDCX 命令，通知 MG2 修改 USER2 的远端媒体信息为 USER1 的媒体属性，同时模式修改为 sendrecv；

54）MG2 回送响应；

55）MG3 检测到用户 User3 的挂机消息，将此挂机事件通过 NTFY 命令上报给 MGC；

56）MGC 回送响应；

57）MGC 向 MG3 发送 DLCX 命令，释放 User3 相关资源；

58）MG3 回送响应；

此时用户 USER1 和用户 USER2 通话；

12.7.4 反极信号

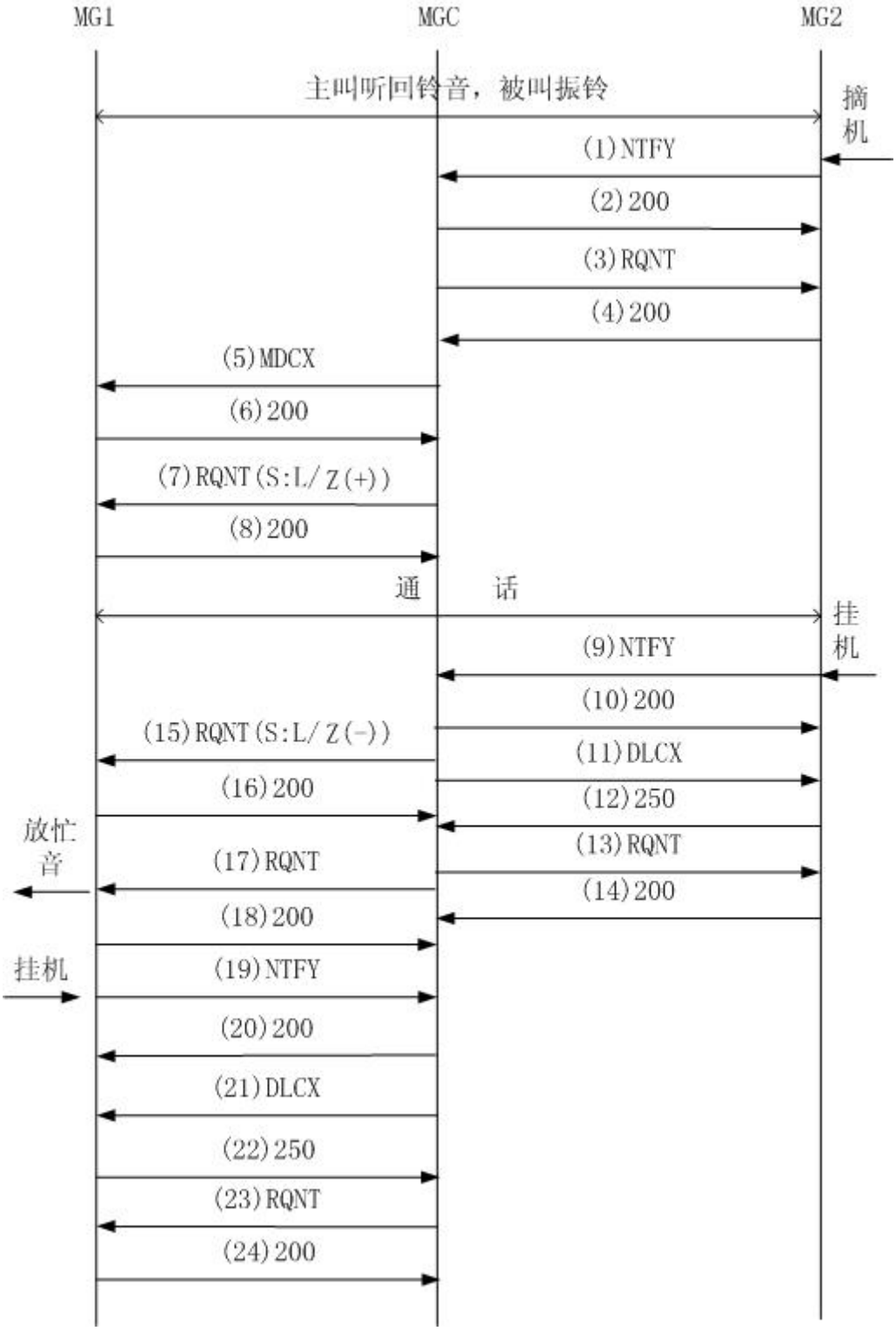


图 12-18：反极性流程图

流程说明：（被叫振铃和主叫听回铃音的过程与 AG 到 AG 的呼叫流程一致）

1） MG2 向 MGC 发送 NTFY 命令，上报被叫用户摘机；

2） MG2 回送响应；

3） MGC 向 MG2 发送 RQNT 命令，让 MG2 检测挂机（L/HU）及拍叉簧（L/HF）；

4） MG2 回送响应;

5） MGC 向 MG1 发送 MDCX 命令，连接模式为 SendRecv，并停回铃音；

6） MG1 回送响应；

7） MGC 向 MG1 发送 RQNT 命令，对该用户进行极性反转，使用 SG：L/Z(+)；

8） MG1 回送响应；主被叫通话；

9） MG2 向 MGC 发送 NTFY 命令，上报用户挂机事件；

10） MGC 回送响应；

11） MGC 向 MG2 发送 DLCX 命令，释放连接资源；

12） MG2 回送响应；

13） MGC 向 MG2 发送 RQNT 命令，要求检测摘机事件（L/HD） ；

14） MG2 回送响应；

15） MGC 向 MG1 发送 RQNT 命令，对该用户进行极性反转，使用 SG：L/Z(-)；

16） MG1 回送响应；

17） MGC 向 MG1 发送 RQNT 命令，对用户放忙音；

18） MG1 回响应；

19） MG1 向 MGC 发送 NTFY 命令，上报用户挂机事件；

20） MGC 回送响应；

21） MGC 向 MG1 发送 DLCX 命令，删除连接；

22） MG1 回送响应

23） MGC 向 MG1 发送 RQNT 命令，检测用户的摘机事件（L/HD）；

24） MG1 回送响应

注：

1. 反极信号还可以采用 L/lsa 。
2. 如果 MG 有检测到挂机或收到放忙音指示即自动做极性反转的功能,对本流程中 MGC 所发的极性反转命令应正常响应并继续完成流程，不应报错。

12.7.5 区别振铃

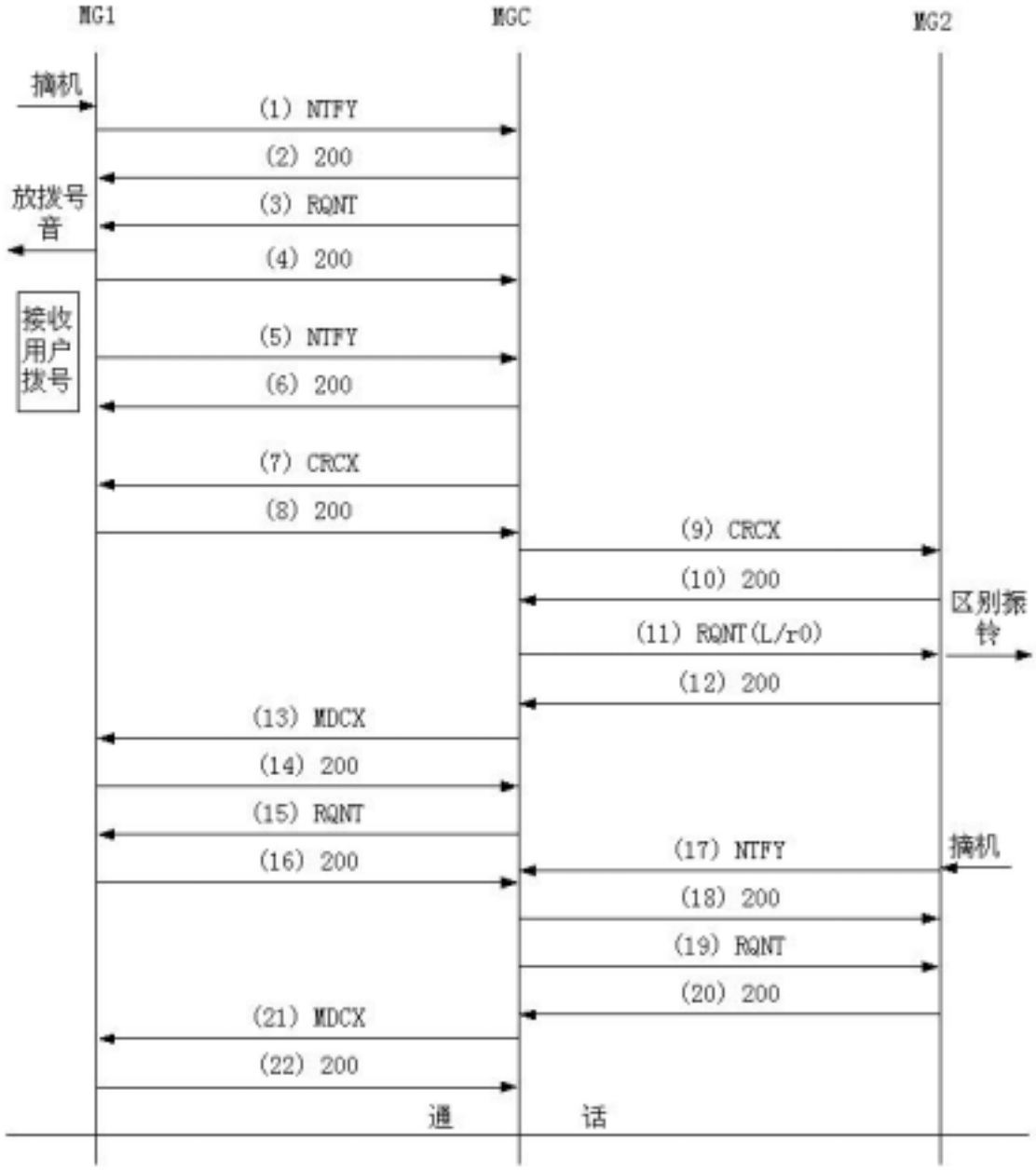


图 12-19：区别振铃流程图

流程说明：

1) MG1 上 User1 摘机，MG1 发送 NTFY（L/HD）命令，通知 MGC；

2) MGC 回响应；

3) MGC 向 MG1 发送 RQNT 命令，送拨号音，下发拨号表并要求监测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

4) MG1 回响应；

5) MG1 发送 NTFY 命令，将用户拨号送给 MGC；

6） MGC 回响应；

7） MGC 向 MG1 发送 CRCX 命令，为主叫创建一个连接，连接模式为 reciveonly；

8） MG1 回响应，并将连接的 SDP 信息返回给 MGC；

9） MGC 向 MG2 发送 CRCX 命令，连接模式为 sendrecv，并且将主叫连接的 SDP 信息带给被叫 MG；

10）MG2 回响应，并将连接的 SDP 信息返回给 MGC；

11）MGC 向 MG2 发送 RQNT 命令，向用户进行区别振铃（可选择振铃方式为：L/r0、L/r1、L/r2、L/r3、L/r4、L/r5、L/r6、 L/r7）；

12）MG2 回响应；

13）MGC 向 MG1 发送 MDCX 命令，把被叫的 SDP 信息带给主叫 MG；

14）MG1 回响应；

15）MGC 向 MG1 发送 RQNT 命令，主叫用户听回铃音；

16）MG1 回响应；

17）被叫用户摘机，MG2 发送 NTFY 命令给 MGC；

18）MGC 回响应；

19）MGC 向 MG2 发送 RQNT 命令，请求被叫 MG 监测挂机(L/HU)及拍叉簧(L/HF)；

20）MG2 回响应；

21）MGC 向 MG1 发送 MDCX 命令，修改连接模式为 sendrecv，并停回铃音；

22）MG1 回响应；主被叫通话；

12.7.6 三方通话

MG1 上的 User1 有三方通话的权利，发起与 MG2 上的 User2 和 MG3 上 User3 进行三方通话；

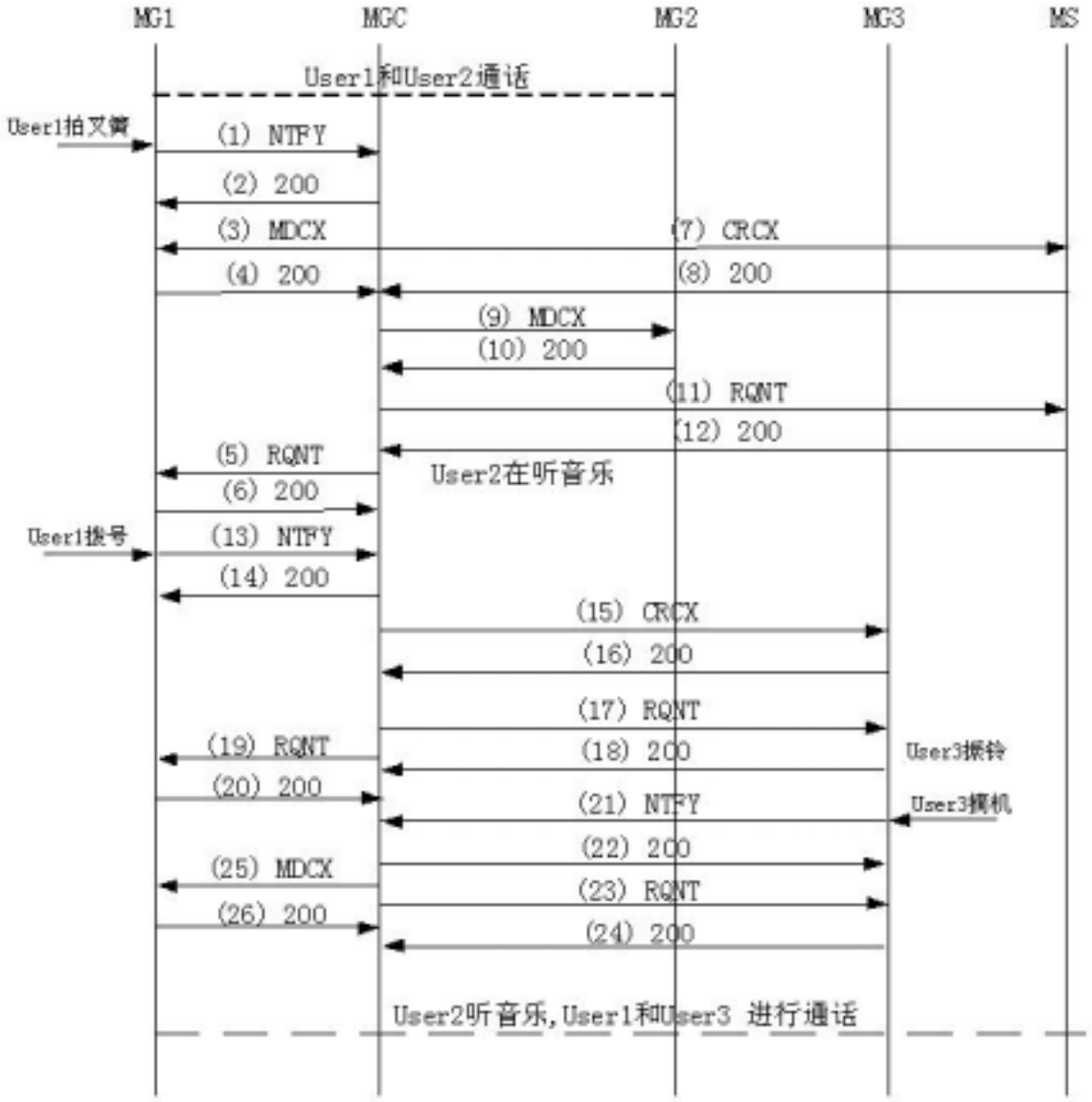


图 12-20：三方通话流程图（1）

流程说明：

User1 和 User2 已处于通话状态，呼叫过程和正常流程相同，省略不描述。

1） User1 拍叉簧。MG1 检测到 User1 的拍叉簧事件，将此事件通过 NTFY 命令上报给 MGC；

2） MGC 向 MG1 返回响应；

3） MGC 向 MG1 发送 MDCX 命令，将其模式修改为 inactive；

4） MG1 向 MGC 返回响应；

5） MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6） MG1 向 MGC 返回响应；

7） MGC 使用 CRCX 命令向 MS 申请资源对 user2 放音乐，附带 User2 的 SDP 信息；

8） MS 向 MGC 返回响应，携带分配的资源终端 Y1 的 SDP 信息；

9） MGC 向 MG2 发送 MDCX 命令修改 User2 的远端 SDP 信息，使其和 Y1 相通，同时修改其模式为 ReciveOnly；

10）MG2 向 MGC 返回响应；

11）MGC 向 MS 发送 RQNT 命令，要求 Y1 向外部发送音乐；

12）MS 向 MGC 返回响应。这时 User2 处于听音乐状态；

13）MG1 上的用户 User1 拨号，MG1 根据 MGC1 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

14）MG1 向 MGC 返回响应；

15）MGC 向 MG3 发送 CRCX 命令，为 User3 分配资源,带 User1 的 SDP 信息；

16）MG3 向 MGC 返回响应,带回 User3 的 SDP 信息；

17）MGC 向 MG3 发送 RQNT 命令，让 MG3 向 User3 振铃；

18）MG3 向 MGC 返回响应；

19）MGC 向 MG1 发送 RQNT 命令，让 MG1 给 User1 送回铃音；

20）MG1 向 MGC 返回响应；

21）MG3 检测到用户 User3 的摘机事件，将此摘机事件通过 NTFY 命令上报给 MGC；

22）MGC 向 MG3 返回响应；

23）MGC 向 MG3 发送 RQNT 命令，让 MG3 检测挂机（L/HU）、拍叉簧（L/HF）

24）MG3 向 MGC 返回响应；

25）MGC 向 MG1 发送 MDCX 消息修改 User1 的远端 SDP 信息，使其和 User3 相通；同时其模式修改为 sendrecv

26）MG1 向 MGC 返回响应；

此时 User2 在听音乐， User1 和 User3 进行通话。

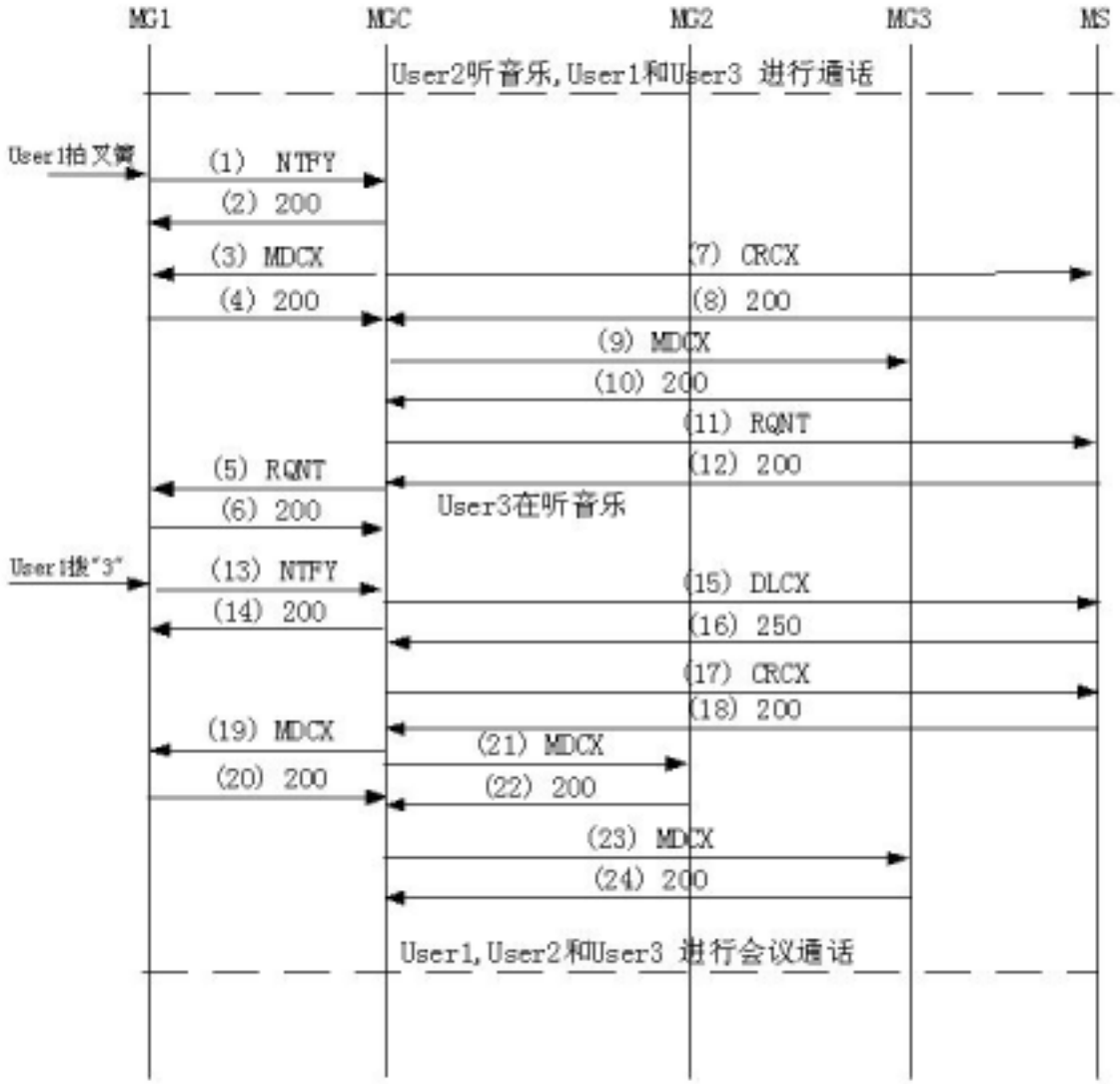


图 12-21：三方通话流程图（2）（召集方拍叉拨 3）

续流程（1）

1） User1 拍叉簧。MG1 检测到 User1 的拍叉簧事件，将此事件通过 NTFY 命令上报给 MGC；

2） MGC 向 MG1 返回响应；

3） MGC 向 MG1 发送 MDCX 命令，将 User1 的模式修改为 inactive；

4） MG1 向 MGC 返回响应；

5） MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6） MG1 向 MGC 返回响应；

7） MGC 使用 CRCX 命令向 MS 申请资源对 user3 放音乐，附带 User3 的 SDP 信息；

8） MS 向 MGC 返回响应，携带分配的资源终端 Y2 的 SDP 信息；

9） MGC 向 MG3 发送 MDCX 命令修改 User3 的远端 SDP 信息，使其和 Y2 相通，同时修改其模式

为 ReciveOnly；

10）MG3 向 MGC 返回响应；

11）MGC 向 MS 发送 RQNT 命令，要求 Y2 向外部发送音乐；

12）MS 向 MGC 返回响应。这时 User3 处于听音乐状态；

13）MG1 上的用户 User1 拨号“ 3 ”，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

14）MG1 向 MGC 返回响应；

15）MGC 向 MS 发送两条 DLCX 命令，要求释放 Y1,Y2,停止对 User2 和 User3 放音，两个 DLCX 命令可以分次发送给 MS，也可一次发送给 MS；

16）MS 向 MGC 返回响应；

17）MGC 使用三个 CRCX 命令向 MS 申请资源，一个申请端点（可以为$，也可以由 MGC 指定）并附带 User1 的 SDP 信息；一个附带 User2 的 SDP 信息；一个附带 User3 的 SDP 信息；三个 CRCX 命令分别向同一个端点建立三个连接，用于会议，三个连接的模式为 confrnce；三个 CRCX 命令可以分次发送给 MS，也可一次发送给 MS；

18）MS 向 MGC 返回响应，携带分配的会议资源终端 C1，C2 和 C3 的 SDP 信息；

19）MGC 向 MG1 发送 MDCX 命令，携带会议资源的 C1 的 SDP 信息。同时设置 User1 的 Mode 为 sendrecv；

20）MG1 向 MGC 返回响应；

21）MGC 向 MG2 发送 MDCX 命令，携带会议资源的 C2 的 SDP 信息。同时设置 User2 的 Mode 为 sendrecv；

22）MG2 向 MGC 返回响应；

23）MGC 向 MG3 发送 MDCX 命令，携带会议资源的 C3 的 SDP 信息。同时设置 User3 的 Mode 为 sendrecv；

24）MG3 向 MGC 返回响应。

User1，User2，User3 进入三方通话状态。

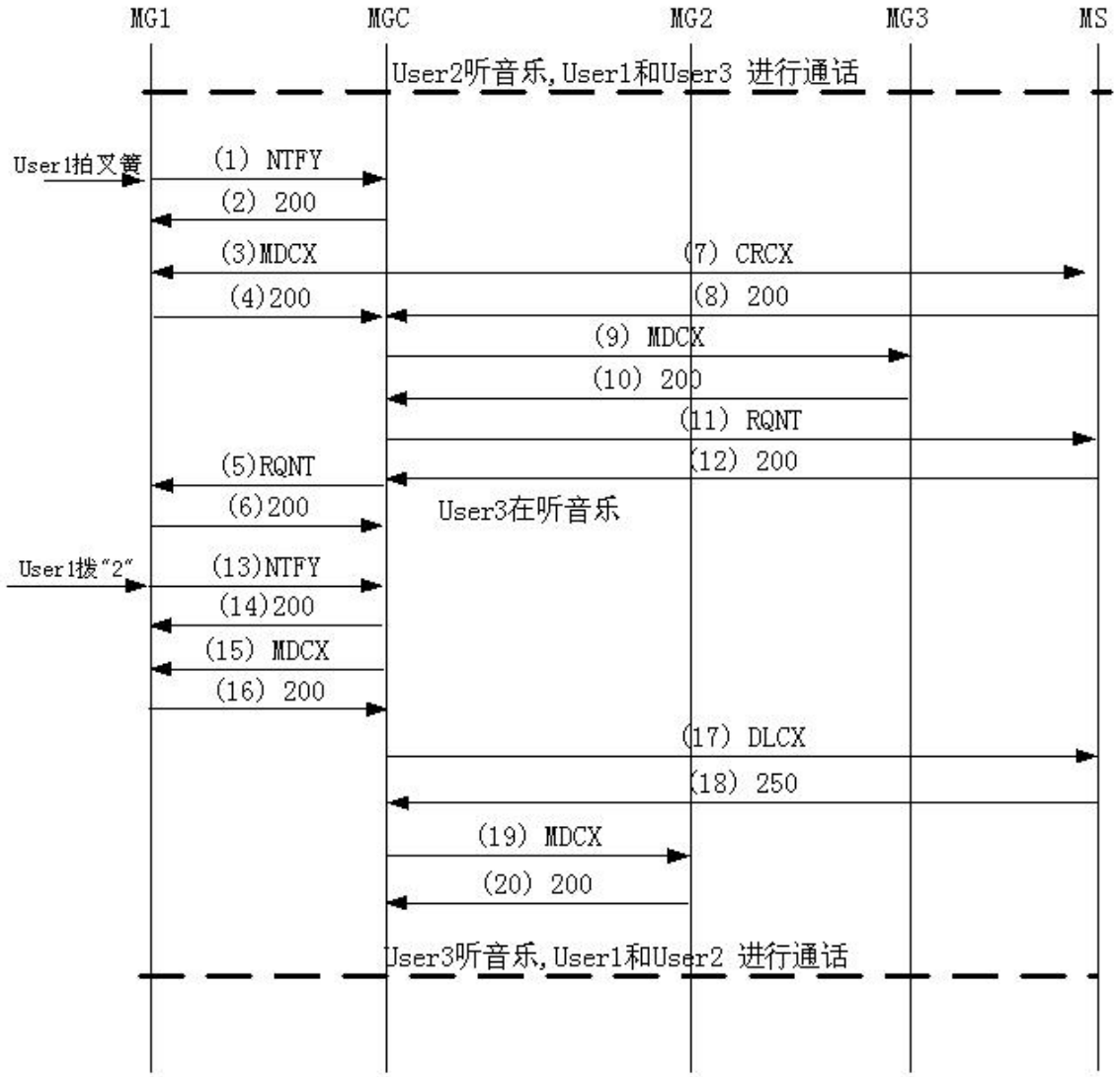


图 12-22：三方通话流程图（3）（召集方拍叉拨 2）

续流程（1）

1） User1 拍叉簧。MG1 检测到 User1 的拍叉簧消息，将此事件通过 NTFY 命令上报给 MGC；

2） MGC 向 MG1 返回响应；

3） MGC 向 MG1 发送 MDCX 命令，将 RTP1 的模式为 inactive；

4） MG1 向 MGC 返回响应；

5） MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6） MG1 向 MGC 返回响应；

7） MGC 使用 CRCX 向 MS 申请资源用来对 user3 放音乐，附带 User3 的 SDP 信息；

8） MS 向 MGC 返回响应，携带分配的资源终端 Y2 的 SDP 信息；

9） MGC 向 MG3 发送 MDCX 修改 User3 的远端 SDP 信息，使其和 Y2 相通，同时修改其模式为 ReciveOnly；

10）MG3 向 MGC 返回响应；

11）MGC 向 MS 发送 RQNT 命令，要求 Y2 向外部发送音乐；

12）MS 向 MGC 返回响应。这时 User3 处于听音乐状态；

13）MG1 上的用户 User1 拨号“ 2 ”，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 上报 MGC；

14）MG1 向 MGC 返回响应；

15）MGC 向 MG1 发送 MDCX 命令修改 User1 的远端 SDP 信息，使其和 User2 相通；同时其模式修改为 sendrecv

16）MG1 向 MGC 返回响应；

17）MGC 向 MS 发送 DLCX 命令，释放 Y1,停止对 User2 放音；

18）MS 向 MGC 返回响应；

19）MGC 向 MG2 发送 MDCX 命令修改 User2 的远端 SDP 信息，使其和 User1 相通，同时其模式改为 sendrecve

20）MG1 向 MGC 返回响应；

此时，User3 听音乐， User1 和 User2 在通话。

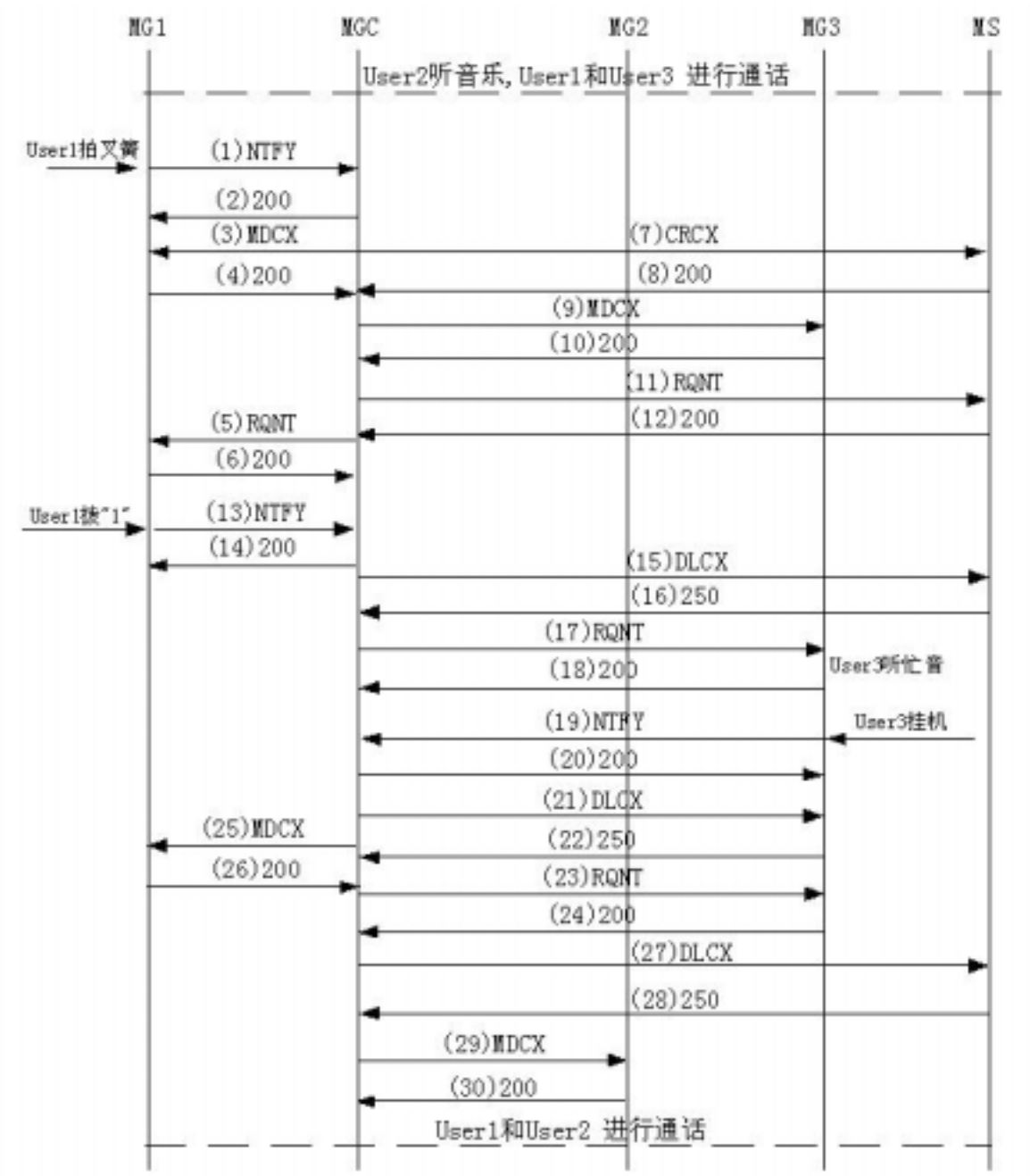


图 12-23：三方通话流程图（4）（召集方拍叉拨 1）

续流程（1）

1） User1 拍叉簧。MG1 检测到 User1 的拍叉簧事件，将此事件通过 NTFY 命令上报给 MGC；

2） MGC 向 MG1 返回响应；

3） MGC 向 MG1 发送 MDCX 命令，将 User1 的模式修改为 inactive；

4） MG1 向 MGC 返回响应；

5） MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6） MG1 向 MGC 返回响应；

7） MGC 使用 CRCX 命令向 MS 申请资源，附带 User3 的 SDP 信息,用来对 user3 放音乐；

8） MS 向 MGC 返回响应，携带分配的资源终端 Y2 的 SDP 信息；

9） MGC 向 MG3 发送 MDCX 命令修改 User3 的远端 SDP 信息，使其和 Y2 相通，同时修改其模式为 ReciveOnly；

10）MG3 向 MGC 返回响应；

11）MGC 向 MS 发送 RQNT 命令，要求 Y2 向外部发送音乐；

12）MS 向 MGC 返回响应。这时 User3 处于听音乐状态；

13）MG1 上的用户 User1 拨号 “1”，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

14）MG1 向 MGC 返回响应；

15）MGC 向 MS 发送 DLCX 命令，释放 Y2,停止对 User3 放音；

16）MS 向 MGC 返回响应；

17）MGC 向 MG3 发送 RQNT 命令,让 MG3 向 User3 放忙音；

18）MG3 向 MGC 返回 Reply；

19）MG3 向 MGC 发送 NTFY 命令，上报 User3 挂机事件；

20）MGC 向 MG3 返回 Reply；

21）MGC 向 MG3 发送 DLCX 命令,释放 User3 资源；

22）MG3 向 MGC 返回响应；

23）MGC 向 MG3 发送 RQNT 命令,让 MG3 对 User3 检测摘机事件；

24）MG3 向 MGC 返回响应；

25）MGC 向 MG1 发送 MDCX 命令修改 User1 的远端 SDP 信息，使其和 User2 相通；同时其模式修改为 sendrecv

26）MG1 向 MGC 返回响应；

27）MGC 向 MS 发送 DLCX 命令，释放 Y1,停止对 User2 放音；

28）MS 向 MGC 返回响应；

29）MGC 向 MG2 发送 MDCX 命令修改 User2 的远端 SDP 信息，使其和 User1 相通；同时其模式修改为 sendrecv

30）MG1 向 MGC 返回响应；

此时，User1 和 User2 通话。

12.7.7 会议电话

MG1 上的 User1 为主席，连续发起 MG2 上的 User2 和 MG3 上 User3 开会。

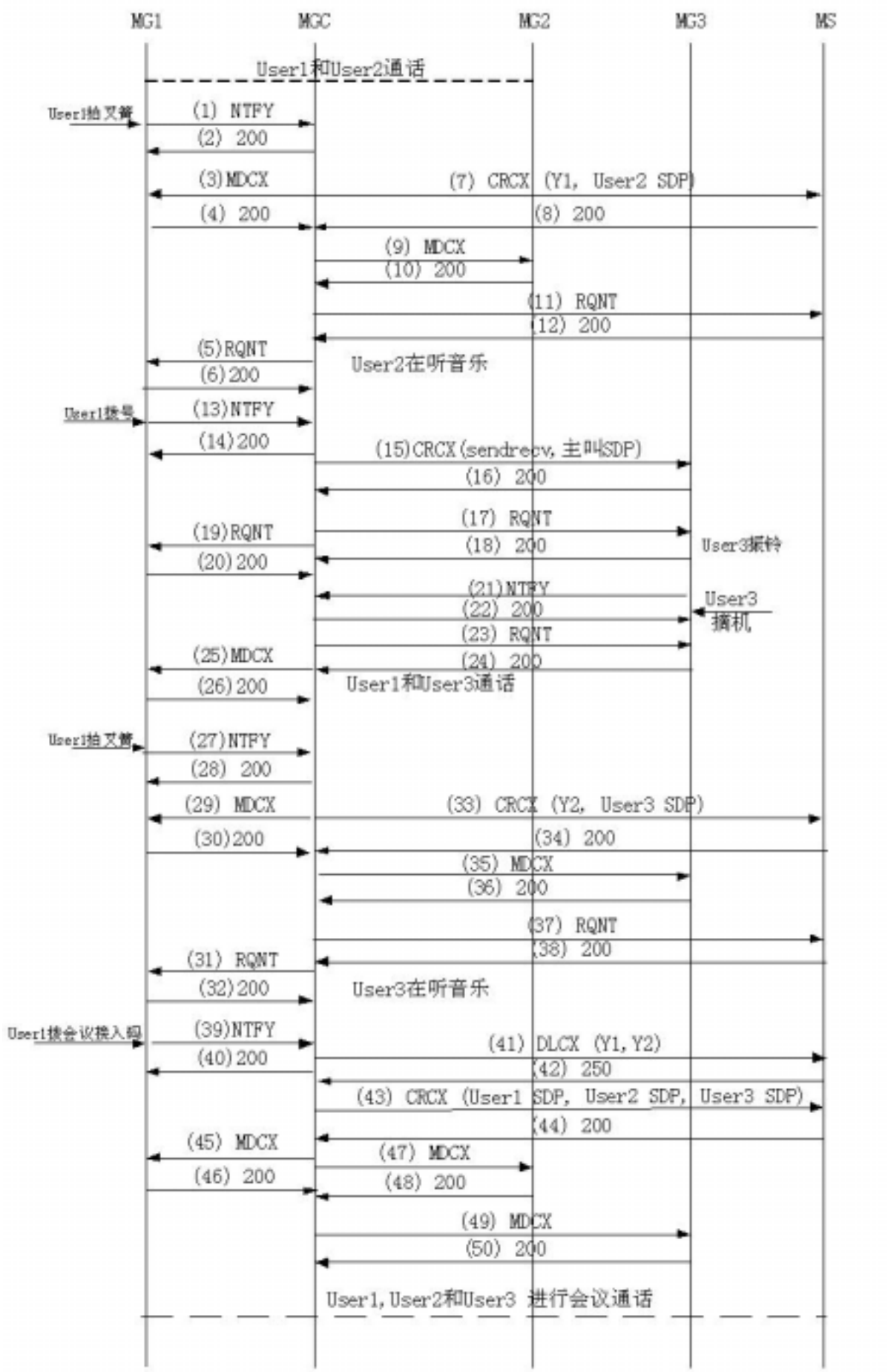

MG1
MGC
MG2
MG3
MS
User1和User2通话
User1拍叉簧
(1) NTFY
(2) 200
(3)MDCX
(7) CRCX (Y1, User2 SDP)
(4) 200
(8) 200
(9) MDCX
(10) 200
(11) RQNT
(12) 200
(5)RQNT
User2在听音乐
(6)200
User1拨号
(13)NTFY
(14)200
(15)CRCX(sendrecv,主叫SDP)
(16) 200
(17) RQNT
(19)RQNT
(18) 200
User3振铃
(20)200
(21)NTFY
User3
摘机
(22) 200
(23) RQNT
(25)MDCX
(24) 200
(26)200
User1和User3通话
User1拍叉簧
(27)NTFY
(28) 200
(29) MDCX
(33) CRCX (Y2, User3 SDP)
(30)200
(34) 200
(35) MDCX
(36) 200
(37) RQNT
(38) 200
(31) RQNT
(32)200
User3在听音乐
User1接会议接入码
(39)NTFY
(41) DLCX (Y1,Y2)
(40)200
(42) 250
(43) CRCX (User1 SDP, User2 SDP, User3 SDP)
(44) 200
(45) MDCX
(47) MDCX
(46) 200
(48) 200
(49) MDCX
(50) 200
User1,User2和User3 进行会议通话

图 12-24： 会议电话流程图

1） MG1 检测到 User1 的拍叉簧消息，将此事件通过 NTFY 命令上报 MGC；

2） MGC 回响应；

3） MGC 向 MG1 发送 MDCX 命令，将 USER1 的模式修改为 inactive；

4） MG1 回响应；

5） MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6） MG1 回送响应；

7） MGC 使用 CRCX 命令向 MS 申请放音资源，附带 USER2 的 SDP 信息；

8） MS 向 MGC 返回响应，携带分配的资源终端 Y1 的 SDP 信息；

9） MGC 向 MG2 发送 MDCX 命令请求修改 USER2 的远端 SDP 信息，使其和 Y1 相通，同时修改其模式为 ReciveOnly；

10） MG2 回送响应；

11） MGC 向 MS 发送 RQNT 命令，要求 Y1 向外部发送音乐；

12） MS 向 MGC 回送响应。这时 User2 处于听音乐状态；

13） MG1 上的用户 User1 拨号，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 命令上报 MGC；

14） MGC 回送响应；

15） MGC 向 MG3 发送 CRCX 命令，连接模式为 sendrecv，带 User1 的 SDP 信息；为 User3 分配资源；

16） MG3 向 MGC 回送响应，并带回 SDP 信息；

17） MGC 向 MG3 发送 RQNT 命令，请求 MG3 给 User3 放振铃音；

18） MG3 回送响应；

19） MGC 向 MG1 发送 RQNT 命令，请求 MG1 给 User1 放回铃音；

20） MG1 回送响应；

21） MG3 检测到用户 User3 的摘机事件，将此摘机事件通过 NTFY 命令上报给 MGC；

22） MGC 回送响应；

23） MGC 向 MG2 发送 RQNT 命令，请求 MG2 监测挂机(L/HU)及拍叉簧(L/HF)；

24） MG2 回响应；

25） MGC 向 MG1 发送 MDCX 命令，修改 User1 的 SDP 信息，使其和 User3 相通；同时模式修改为 SendReceive

26） MG1 回送响应；此时 User1 和 User3 通话， User2 在听音乐；

27）MG1 检测到 User1 的拍叉簧事件，将此事件通过 NTFY 命令上报给 MGC；

28）MGC 回送响应；

29）MGC 向 MG1 发送 MDCX 命令，将其用户 User1 的模式修改为 inactive；

30）MG1 回送响应；

31）MGC 向 MG1 发送 RQNT 命令，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

32）MG1 回送响应；

33）MGC 使用 CRCX 命令向 MS 申请放音资源，附带 User3 的 SDP 信息；

34）MS 向 MGC 回送响应，携带分配的资源终端 Y2 的 SDP 信息；

35）MGC 向 MG3 发送 MDCX 命令修改 User3 的 SDP 信息，使其和 Y2 相通，同时修改其模式为 ReciveOnly；

36）MG3 回送响应；

37）MGC 向 MS 发送 RQNT 命令，请求 Y2 向外部发送音乐；

38）MS 向 MGC 回送响应。这时 User3 处于听音乐状态；

39）MG1 上的用户 User1 拨会议接入号，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 消息上报 MGC；

40）MG1 回送响应；

41）MGC 向 MS 发送两条 DLCX 命令，要求释放 Y1、Y2,停止对 User2 和 User3 放音，两个 DLCX 命令可以分次发送给 MS，也可一次发送给 MS；

42）MS 回送响应；

43）MGC 使用三个 CRCX 命令向 MS 申请资源，一个申请端点（可以为$，也可以由 MGC 指定）并附带 User1 的 SDP 信息；一个附带 User2 的 SDP 信息；一个附带 User3 的 SDP 信息；三个 CRCX 命令分别向同一个端点建立三个连接，用于会议，三个连接的模式为 confrnce；三个 CRCX 命令可以分次发送给 MS，也可一次发送给 MS；

44）MS 向 MGC 回送响应，携带分配的会议资源终端 C1，C2 和 C3 的 SDP 信息；

45）MGC 向 MG1 发送 MDCX 命令，携带会议资源的 C1 的 SDP 信息。并设置其用户 User1 的 Mode 为 sendrecv；

46）MG1 回送响应；

47）MGC 向 MG2 发送 MDCX 命令，携带会议资源的 C2 的 SDP 信息。并设置其用户 User2 的 Mode 为 sendrecv；

48）MG2 回送响应；

49）MGC 向 MG3 发送 MDCX 命令，携带会议资源的 C3 的 SDP 信息。并设置其用户 User3 的 Mode 为 sendrecv；

50）MG3 回送响应；

此时，User1、 User2 和 User3 处于会议呼叫。

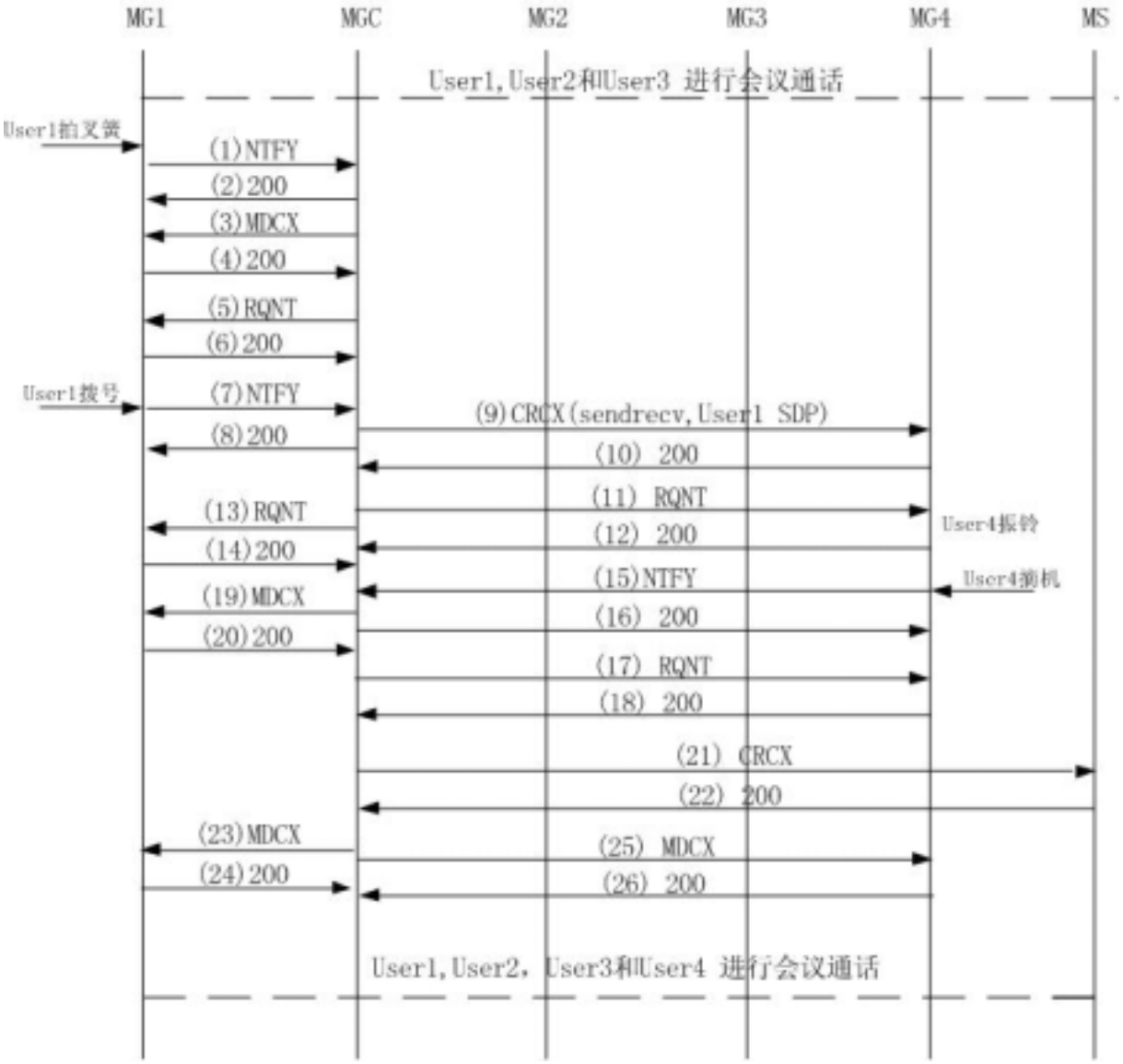


图 12-25： 会议电话主席拍叉簧叫入第四方

续前一流程：

1) MG1 检测到 User1 的拍叉簧事件，将此事件通过 NTFY 命令上报给 MGC；

2) MGC 回送响应；

3) MGC 向 MG1 发送 MDCX 消息，将 User1 的模式修改为 inactive；

4) MG1 回送响应；

5) MGC 向 MG1 发送 RQNT 消息，请求 MG1 放拨号音，下发拨号表并要求检测用户拨号、挂机（L/HU）、拍叉簧（L/HF）及放音结束事件（L/OC）；

6) MG1 回送响应；

7) MG1 上的用户 User1 拨号，MG1 根据 MGC 所下发的号码表进行收号，并将所拨号码及匹配结果用 NTFY 消息上报 MGC；

8） MGC 回送响应；

9） MGC 向 MG4 发送 CRCX 消息，为 User4 分配资源，模式设为 sendrecv，并附带 User1 的 SDP 信息；

10）MG4 向 MGC 回送响应，并带回为 User4 分配的 SDP 信息；

11）MGC 向 MG4 发送 RQNT 命令，让 MG4 对 User4 振铃；

12）MG4 回送响应；

13）MGC 向 MG1 发送 RQNT 命令，让 MG1 给 User1 放回铃音；

14）MG1 回送响应；

15）MG4 检测到用户 User4 的摘机，将此摘机事件通过 NTFY 命令上报给 MGC；

16）MGC 回送响应；

17）MGC 向 MG4 发送 RQNT 命令，请求 MG4 检测 User4 挂机、拍叉簧事件；

18）MG4 向 MGC 回送响应；

19）MGC 向 MG1 发送 MDCX 命令，将 User1 的模式修改为 sendrecv；

20）MG1 回送响应；

到此，user1 和 user4 可以通话后，再由 user1 拍叉进入会议（该过程省略）；也可以直接执行以下步骤进入会议。

21）MGC 用 CRCX 命令向 MS 申请会议资源，向前三方所连的 endpoint 增加一个连接，用于 User4 加入会议，模式为 confrnce，并附带远端为 User4 的 SDP；

22）MS 向 MGC 回送响应，携带分配的会议资源 C4 的 SDP 信息；

23）MGC 向 MG1 发 MDCX 消息，修改远端 SDP 为 C1；

24）MG1 回送响应；

25）MGC 向 MG4 发 MDCX 消息，修改远端 SDP 为 C4；

26）MG4 回送响应。

此时，User1、 User2、 User3 和 User4 处于会议呼叫。

12.8 T.38传真业务流程

网关应该支持 T.38 传真编码算法，在软交换的控制下能够完成 T.38 方式的传真业务。采用 G.711 语音编码算法传送传真数据的方式可以作为 T.38 方式的补充，在网关的 T.38 编码资源耗尽时采用。

12.8.1 AG-AG传真建立（结束后切回语音）

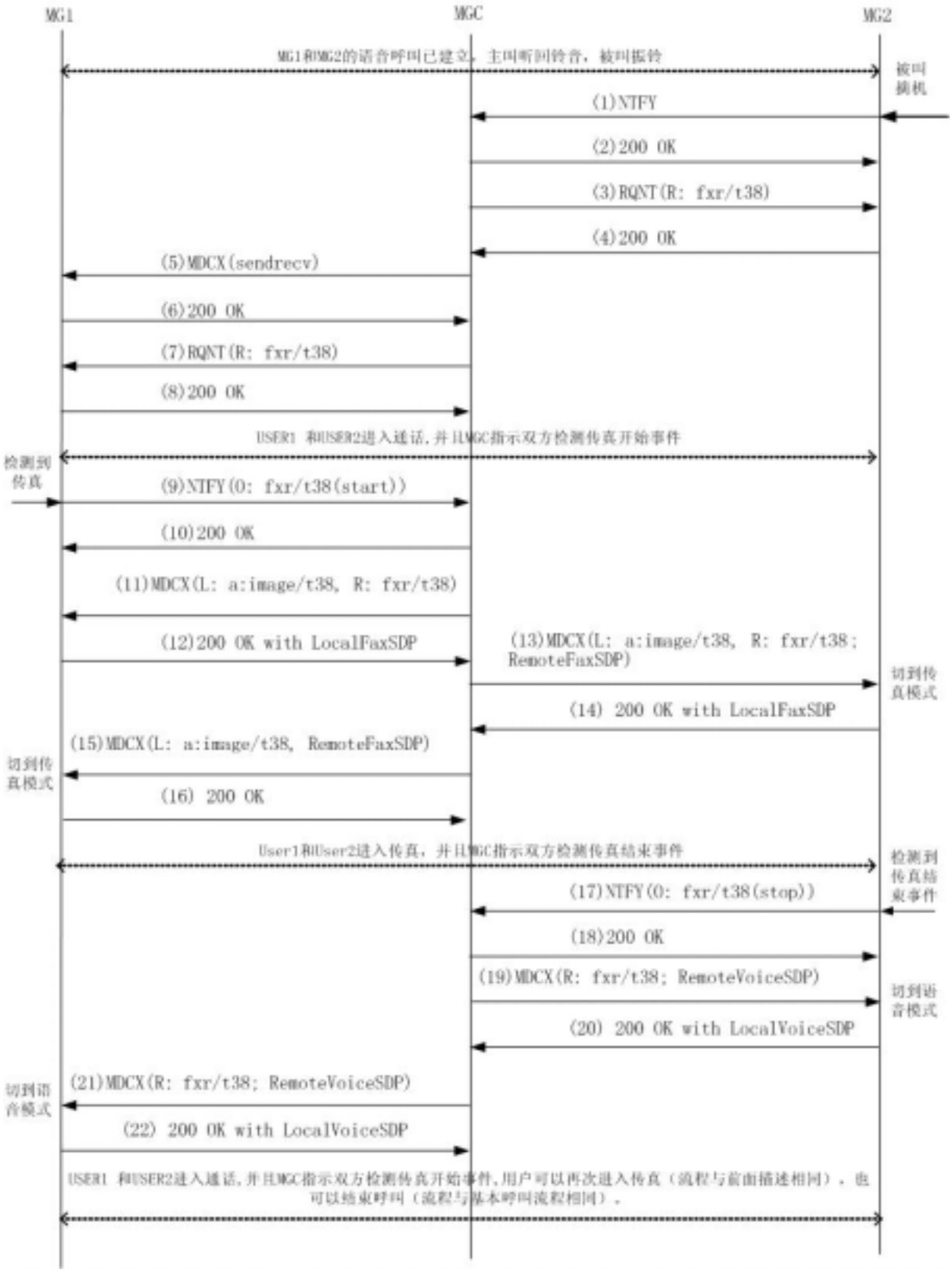


图 12-26： AG-AG 传真建立流程

流程说明：

MG1 上的用户 User1 为主叫，MG2 上的用户 User2 为被叫，User1 与 User2 间的语音呼叫建立流程参见本标准的 12.3，本流程从被叫摘机开始来说明。

1） User2 振铃后 MG2 检测到用户 User2 摘机，将此摘机事件通过 NTFY 命令上报给 MGC；

2） MGC 回送响应；

3） MGC 向 MG2 发送 RQNT 消息，请求 MG2 检测挂机（L/hu）、拍叉簧（L/hf）、传真开始（FXR/t38）事件；

4） MG2 回送响应；

5） MGC 向 MG1 发送 MDCX 命令，修改连接模式为 sendrecv；

6） MG1 回送响应；

7） MGC 向 MG1 发送 RQNT 消息，请求 MG1 检测挂机（L/hu）、拍叉簧（L/hf）、传真开始（FXR/t38）事件；

8） MG1 回送响应。至此，MG1 上的 User1 与 MG2 上的 User2 的语音呼叫建立完毕，并且都在检测传真开始事件；

9） MG1 检测到用户 User1 的 V.21 传真开始事件，将此传真开始事件 FXR/t38(start)通过 NTFY 命令上报给 MGC；

10） MGC 回送响应；

11） MGC 向 MG1 发送 MDCX 命令，请求 MG1 检测挂机（L/hu）、拍叉簧（L/hf）、传真结束（FXR/t38）事件；同时用本地连接选项参数（L）询问 RTP1 的本端传真媒体的 IP 地址及端口号、传真算法等；

12） MG1 回送响应，其中携带 MG1 的传真能力；

13） MGC 向 MG2 发送 MDCX 命令，用本地连接选项参数（L）询问 RTP2 的本端传真媒体的 IP 地址、端口号及传真算法等，并指示网关要切换到“ image/t38 ” MIME 类型；同时 MGC 告知 RTP2 的远端传真媒体的 IP 地址及端口号、传真算法等；

14） MG2 确定一种传真算法，优先 T.38，在响应中将传真媒体信息带给 MGC；

15） MGC 向 MG1 发送 MDCX 消息，设置 RTP1 的远端传真媒体 IP 地址及端口号、传真算法等，并指示网关切换到“ image/t38 ” MIME 类型；

16） MG1 回送响应，至此，MG1 的 User1 与 MG2 的 User2 的传真呼叫建立完毕，并且都在检测传真结束事件；

17） 假设 MG2 先检测到用户 User2 的传真结束事件，将此传真结束事件 FXR/t38(stop)通过 NTFY 命令上报给 MGC；

18） MGC 回送响应；

19） MGC 向 MG2 发送 MDCX 命令，请求 MG2 检测挂机（L/hu）、拍叉簧（L/hf）、传真开始（FXR/t38）事件；并设置 RTP2 的本端及远端语音 RTP 地址及端口号、语音压缩算法等；

20） MG2 回送响应；

21） MGC 向 MG1 发送 MDCX 命令，请求 MG2 检测挂机（L/hu）、拍叉簧（L/hf）、传真开始（FXR/t38）事件；并设置 RTP1 的本端及远端语音 RTP 地址及端口号、语音压缩算法等；

22）MG1 向 MGC 回送应答，至此 MG1 的 User1 与 MG2 的 User2 之间又恢复成语音呼叫，并且都在检测传真开始事件；用户可以选择继续语音呼叫；也可以选择再次进入传真呼叫（流程同前面描述）；用户也可以选择结束呼叫（呼叫释放流程参见本标准的 12.4）。

补充说明：

i. 如果在网关上报传真结束事件后，用户很快就挂机了，则不再切回语音模式，而直接进入呼叫释放流程；如果在上报传真结束事件前用户挂机，则也直接进入呼叫释放流程。

ii. 如果后检测到用户传真结束事件的 MG1 将传真结束事件（FXR/t38(stop)）通过 NTFY 命令上报给 MGC，MGC 必须向 MG1 返回正常响应。

iii. 信令流程参考如下：

```
序号 1 MG2->MGC

   NTFY 1 aaln/0@gw2.whatever.net MGCP 1.0

   X: 1

   O: L/hd


序号 2 MGC->MG2

   200 1 OK


序号 3 MGC->MG2

   RQNT 100 aaln/0@gw2.whatever.net MGCP 1.0

   X: 2

   R: L/hu(N), L/hf(N), FXR/t38


序号 4 MG2->MGC

   200 100  OK


序号 5 MGC->MG1

   MDCX 101 aaln/0@gw1.whatever.net MGCP 1.0

   C: 1

   I: 1

   M: sendrecv
```

序号 6 MG1->MGC

```
200 101  OK
```

序号 7 MGC->MG1

```
RQNT 102 aaln/0@gw1.whatever.net MGCP 1.0
X: 3
R: L/hu(N), L/hf(N), FXR/t38
```

序号 8 MG1->MGC

```
200 102  OK
```

序号 9 MG1->MGC

```
NTFY 2 aaln/0@gw1.whatever.net MGCP 1.0
X: 3
O: FXR/t38(start)
```

序号 10 MGC->MG1

```
200 2 OK
```

序号 11 MGC->MG1

```
MDCX 103 aaln/0@gw1.whatever.net MGCP 1.0
C: 1
I: 1
L: e:off, a:image/t38;PCMU
X: 4
R: L/hu(N), L/hf(N), FXR/t38
```

序号 12 MG1->MGC

```
200 103  OK
```

```
v=0
o=- 25678 753850 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=image  1296 udptl t38
v=0
o=- 25678 753850 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio  1296 RTP/AVP 0
```

序号 13 MGC->MG2

```
MDCX 104 aaln/0@gw2.whatever.net MGCP 1.0
L: e:off, a:image/t38;PCMU
C: 2
I: 2

v=0
o=- 25678 753850 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=image  1296 udptl t38
v=0
o=- 25678 753850 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
```

m=audio 1296 RTP/AVP 0

序号 14 MG2->MGC

200 104 OK

v=0

o=- 25678 753850 IN IP4 192.168.1.2

s=-

c=IN IP4 192.168.1.2

t=0 0

m=image 3456 udptl t38

序号 15 MGC->MG1

MDCX 105 aaln/0@gw1.whatever.net MGCP 1.0

L: e:off, a:image/t38;

C: 1

I: 1

v=0

o=- 25678 753850 IN IP4 192.168.1.2

s=-

c=IN IP4 192.168.1.2

t=0 0

m=image 3456 udptl t38

序号 16 MG1->MGC

200 105 OK

序号 17 MG2->MGC

NTFY 3 aaln/0@gw2.whatever.net MGCP 1.0

X: 2

O: FXR/t38(stop)

序号 18 MG2->MGC

200 3 OK

序号 19 MGC->MG2

MDCX 106 aaln/0@gw2.whatever.net MGCP 1.0

C: 2

I: 2

L: e:on, p:20, a:PCMU;PCMA

X: 5

R: L/hu(N), L/hf(N), FXR/t38

v=0

o=- 25678 753850 IN IP4 192.168.1.1

s=-

c=IN IP4 192.168.1.1

t=0 0

m=audio 1296 RTP/AVP 0

序号 20 MG2->MGC

200 106 OK

v=0

o=- 25678 753850 IN IP4 192.168.1.2

s=-

c=IN IP4 192.168.1.2

t=0 0

m=audio 3456 RTP/AVP 0

序号 21 MGC->MG1

```
MDCX 107 aaln/0@gw1.whatever.net MGCP 1.0
C: 1
I: 1
L: e:on, p:20, a:PCMU;PCMA
X: 6
R: L/hu(N), L/hf(N), FXR/t38

v=0
o=- 25678 753850 IN IP4 192.168.1.2
s=-
c=IN IP4 192.168.1.2
t=0 0
m=audio  3456 RTP/AVP 0
```

序号 22 MG1->MGC

```
200 107  OK

v=0
o=- 25678 753850 IN IP4 192.168.1.1
s=-
c=IN IP4 192.168.1.1
t=0 0
m=audio  1296 RTP/AVP 0
```

12.8.2 TG-TG传真建立（结束后切回语音）

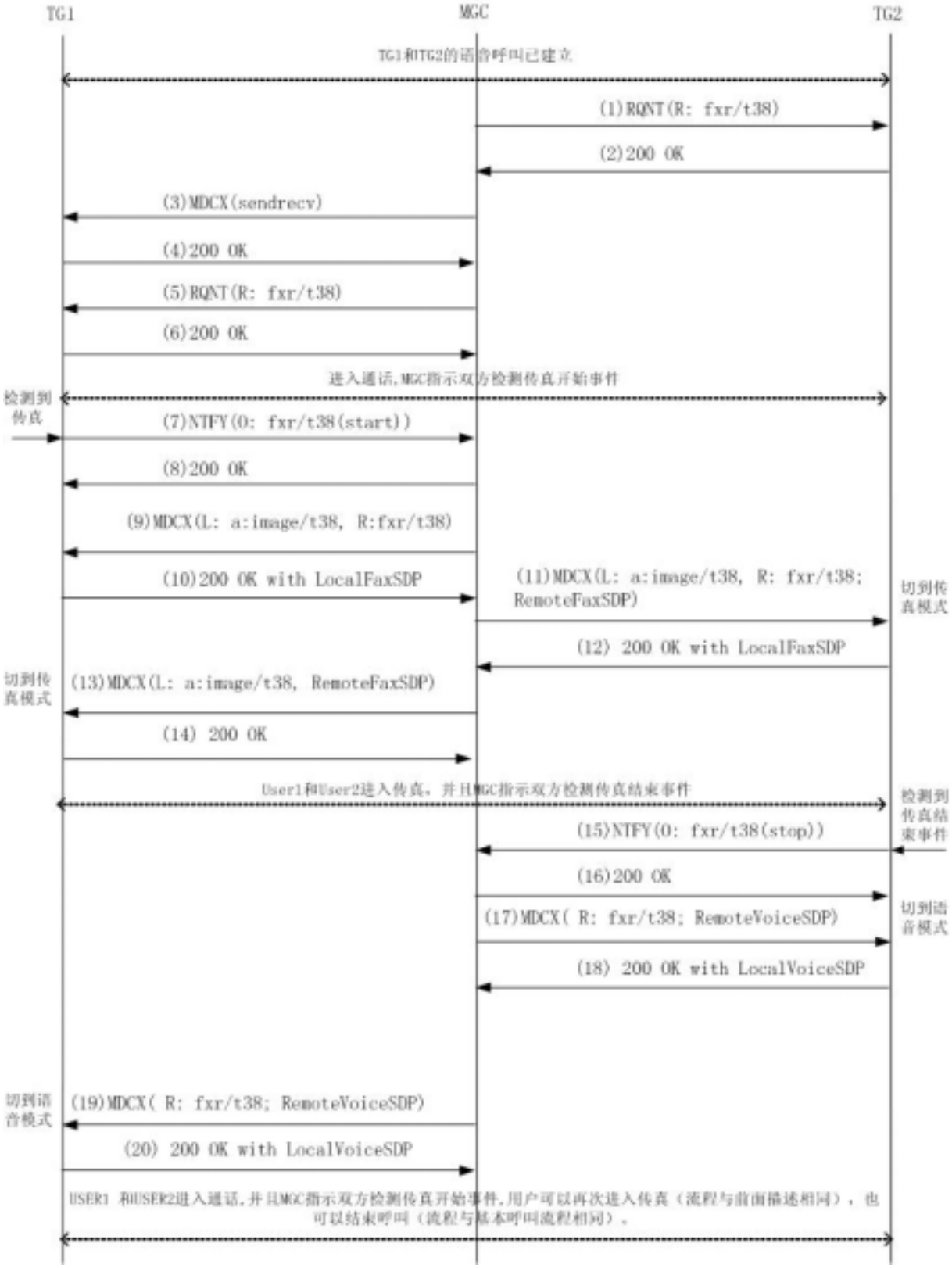


图 12-27： TG-TG 传真建立流程

流程说明：

TG1 和 TG2 之间的语音呼叫已建立，TG1 与 TG2 间的语音呼叫建立流程参见本标准的 14.4，本流程从被叫摘机开始来说明。

1） MGC 向 TG2 发送 RQNT 消息，请求 TG2 检测传真开始（FXR/t38）事件；

2） TG2 回送响应；

3） MGC 向 TG1 发送 MDCX 命令，修改连接模式为 sendrecv；

4） TG1 回送响应；

5） MGC 向 TG1 发送 RQNT 消息，请求 TG1 检测传真开始（FXR/t38）事件；

6） TG1 回送响应。至此，TG1 与 TG2 的语音呼叫建立完毕，并且都在检测传真开始事件；

7） TG1 检测到 V.21 传真开始事件，将此传真开始事件 FXR/t38(start)通过 NTFY 命令上报给 MGC；

8） MGC 回送响应；

9） MGC 向 TG1 发送 MDCX 命令，请求 TG1 检测传真结束（FXR/t38）事件；同时用本地连接选项参数（L）询问 TG1 的本端传真媒体的 IP 地址及端口号、传真算法等；

10） TG1 回送响应，其中携带 TG1 的传真能力；

11） MGC 向 TG2 发送 MDCX 命令，用本地连接选项参数（L）询问 RTP2 的本端传真媒体的 IP 地址及端口号、传真算法等，并指示网关切换到“ image/t38 ” MIME 类型；同时告知 TG2 的远端传真媒体的 IP 地址及端口号、传真算法等；

12） TG2 确定一种传真算法，优先 T.38，在响应中将传真媒体信息带给 MGC；

13） MGC 向 TG1 发送 MDCX 消息，设置 TG1 的远端传真媒体 IP 地址及端口号、传真算法等，同时在本地连接选项参数（L）中指示网关切换到“ image/t38 ” MIME 类型；

14） TG1 回送响应，至此，TG1 与 TG2 的传真呼叫建立完毕，并且都在检测传真结束事件；

15） 假设 TG2 先检测到传真结束事件，将此传真结束事件 FXR/t38(stop)通过 NTFY 命令上报给 MGC；

16） MGC 回送响应；

17） MGC 向 TG2 发送 MDCX 命令，请求 TG2 检测传真开始（FXR/t38）事件；并设置 TG2 的本端及远端语音 RTP 地址及端口号、语音压缩算法等；

18） TG2 回送响应；

19） MGC 向 TG1 发送 MDCX 命令，请求 TG2 传真开始（FXR/t38）事件；并设置 TG1 的本端及远端语音 RTP 地址及端口号、语音压缩算法等；

20） TG1 向 MGC 回送应答，至此 TG1 与 TG2 之间又恢复成语音呼叫，并且都在检测传真开始事件；用户可以选择继续语音呼叫；也可以选择再次进入传真呼叫（流程同前面描述）；用户也可以选择结束呼叫（呼叫释放流程参见本标准的 12.4）。

补充说明：

如果后检测到传真结束事件的 TG1 将传真结束事件(FXR/t38(stop))通过 NTFY 命令上报给 MGC，MGC 必须向 TG1 返回正常响应。

13 目前常用包及扩展包

目前常用包及扩展包如下：

Package	Name
Generic Media Package	G
DTMF package	D
MF Package	M
Trunk Package	T
Line Package	L
Handset Package	H
RTP Package	R
Network Access Server Package	N
Announcement Server Package	A
Advanced Audio Server Base Package	AASB
Advanced Audio Server Set Package	AASS
Advanced Audio Server Conformance Package	AASC
Script Package	Script
Base package	B

附录 A：协议的正式语法描述

本部分提供正式的协议语法描述，采用在 RFC2234 中定义的“扩展的 BNF 语法规则”。该语法利用了定义在 RFC2234 第 6.1 章中的核心规则，此处并没有包含该规则。此外，语法遵从 RFC2234 中定义的大小写敏感性约定，也就是说，MGCP 是与大小写无关的（但 SDP 不是）。应该注意，ABNF 没有提供隐式的线性空白符（linear white space）规则，因此，MGCP 消息必须遵循本语法提供的显式的线性空白符规则。然而，符合一般的鲁棒性原则，强烈主张实施者容忍收到的消息中的额外的线性空白符。

```
MGCPMessage = MGCPCommand / MGCPResponse

MGCPCommand = MGCPCommandLine 0*(MGCPParameter) [EOL *SDPinformation]
```

```
MGCPCommandLine = MGCPVerb 1*(WSP) transaction-id 1*(WSP)
                        endpointName 1*(WSP) MGCPversion EOL

MGCPVerb = "EPCF" / "CRCX" / "MDCX" / "DLCX" / "RQNT"
         / "NTFY" / "AUEP" / "AUCX" / "RSIP" / extensionVerb

extensionVerb  = ALPHA 3(ALPHA / DIGIT) ; experimental starts with X

transaction-id = 1*9(DIGIT)

endpointName      = LocalEndpointName "@" DomainName
LocalEndpointName = LocalNamePart 0*("/" LocalNamePart)
LocalNamePart     = AnyName / AllName / NameString
AnyName           = "$"
AllName           = "*"
NameString        = 1*(range-of-allowed-characters)
; VCHAR except "$", "*", "/", "@"
range-of-allowed-characters  = %x21-23 / %x25-29 / %x2B-2E
                             / %x30-3F / %x41-7E

DomainName = 1*255(ALPHA / DIGIT / "." / "-")    ; as defined
           / "#" number                          ; in RFC 821
           / "[" IPv4address / IPv6address "]"   ; see RFC 2373

; Rewritten to ABNF from RFC 821
number =  1*DIGIT

;From RFC 2373
IPv6address = hexpart [ ":" IPv4address ]
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT

; this production, while occurring in RFC2373, is not referenced
; IPv6prefix  = hexpart "/" 1*2DIGIT
hexpart = hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
hexseq  = hex4 *( ":" hex4)
hex4    = 1*4HEXDIG

MGCPversion = "MGCP" 1*(WSP) 1*(DIGIT) "." 1*(DIGIT)
                           [1*(WSP) ProfileName]
ProfileName = VCHAR *( WSP / VCHAR)

MGCPParameter = ParameterValue EOL

; Check infoCode if more parameter values defined
```

```
; Most optional values can only be omitted when auditing
ParameterValue = ("K"  ":" 0*(WSP)  [ResponseAck])
               / ("B"  ":" 0*(WSP)  [BearerInformation])
               / ("C"  ":" 0*(WSP)  CallId)
               / ("I"  ":" 0*(WSP)  [ConnectionId])
               / ("N"  ":" 0*(WSP)  [NotifiedEntity])
               / ("X"  ":" 0*(WSP)  [RequestIdentifier])
               / ("L"  ":" 0*(WSP)  [LocalConnectionOptions])
               / ("M"  ":" 0*(WSP)  ConnectionMode)
               / ("R"  ":" 0*(WSP)  [RequestedEvents])
               / ("S"  ":" 0*(WSP)  [SignalRequests])
               / ("D"  ":" 0*(WSP)  [DigitMap])
               / ("O"  ":" 0*(WSP)  [ObservedEvents])
               / ("P"  ":" 0*(WSP)  [ConnectionParameters])
               / ("E"  ":" 0*(WSP)  ReasonCode)
               / ("Z"  ":" 0*(WSP)  [SpecificEndpointID])
               / ("Z2" ":" 0*(WSP)  SecondEndpointID)
               / ("I2" ":" 0*(WSP)  SecondConnectionID)
               / ("F"  ":" 0*(WSP)  [RequestedInfo])
               / ("Q"  ":" 0*(WSP)  QuarantineHandling)
               / ("T"  ":" 0*(WSP)  [DetectEvents])
               / ("RM" ":" 0*(WSP)  RestartMethod)
               / ("RD" ":" 0*(WSP)  RestartDelay)
               / ("A"  ":" 0*(WSP)  [Capabilities])
               / ("ES" ":" 0*(WSP)  [EventStates])
               / ("PL" ":" 0*(WSP)  [PackageList])    ; Auditing only
               / ("MD" ":" 0*(WSP)  MaxMGCPDatagram)  ; Auditing only
               / (extensionParameter ":" 0*(WSP) [parameterString])

; A final response may include an empty ResponseAck
ResponseAck =  confirmedTransactionIdRange
               *( ","  0*(WSP) confirmedTransactionIdRange )

confirmedTransactionIdRange = transaction-id ["-" transaction-id]

BearerInformation = BearerAttribute 0*("," 0*(WSP) BearerAttribute)
BearerAttribute   = ("e" ":" BearerEncoding)
                  / (BearerExtensionName [":" BearerExtensionValue])
BearerExtensionName  = PackageLCOExtensionName
BearerExtensionValue = LocalOptionExtensionValue
BearerEncoding = "A" / "mu"

CallId = 1*32(HEXDIG)
```

```
; The audit request response may include a list of identifiers
ConnectionId = 1*32(HEXDIG) 0*("," 0*(WSP) 1*32(HEXDIG))
SecondConnectionID = ConnectionId

NotifiedEntity = [LocalName "@"] DomainName [":" portNumber]
LocalName  = LocalEndpointName            ; No internal structure

portNumber = 1*5(DIGIT)

RequestIdentifier = 1*32(HEXDIG)

LocalConnectionOptions = LocalOptionValue 0*(WSP)
                          0*("," 0*(WSP) LocalOptionValue 0*(WSP))
LocalOptionValue = ("p"  ":" packetizationPeriod)
                 / ("a"  ":" compressionAlgorithm)
                 / ("b"  ":" bandwidth)
                 / ("e"  ":" echoCancellation)
                 / ("gc" ":" gainControl)
                 / ("s"  ":" silenceSuppression)
                 / ("t"  ":" typeOfService)
                 / ("r"  ":" resourceReservation)
                 / ("k"  ":" encryptiondata)
                 / ("nt" ":" ( typeOfNetwork /
                                    supportedTypeOfNetwork))
                 / (LocalOptionExtensionName
                        [":" LocalOptionExtensionValue])

Capabilities    =  CapabilityValue 0*(WSP)
                     0*("," 0*(WSP) CapabilityValue 0*(WSP))
CapabilityValue = LocalOptionValue
                / ("v" ":" supportedPackages)
                / ("m" ":" supportedModes)

PackageList     = pkgNameAndVers 0*("," pkgNameAndVers)
pkgNameAndVers  = packageName ":" packageVersion
packageVersion  = 1*(DIGIT)

packetizationPeriod  = 1*4(DIGIT) ["-" 1*4(DIGIT)]
compressionAlgorithm = algorithmName 0*(";" algorithmName)

algorithmName        = 1*(SuitableLCOCharacter)
bandwidth            = 1*4(DIGIT) ["-" 1*4(DIGIT)]
echoCancellation     = "on" / "off"
gainControl          = "auto" / ["-"] 1*4(DIGIT)
```

```
silenceSuppression   = "on" / "off"
typeOfService        = 1*2(HEXDIG)     ; 1 hex only for capabilities
resourceReservation  = "g" / "cl" / "be"

;encryption parameters are coded as in SDP (RFC 2327)
;NOTE: encryption key may contain an algorithm as specified in RFC 1890
encryptiondata = ( "clear" ":" encryptionKey )
               / ( "base64" ":" encodedEncryptionKey )
               / ( "uri" ":" URItoObtainKey )
               / ( "prompt" ) ; defined in SDP, not usable in MGCP!

encryptionKey = 1*(SuitableLCOCharacter) / quotedString
; See RFC 2045
encodedEncryptionKey = 1*(ALPHA / DIGIT / "+" / "/" / "=")
URItoObtainKey = 1*(SuitableLCOCharacter) / quotedString

typeOfNetwork = "IN" / "ATM" / "LOCAL" / OtherTypeOfNetwork
; Registered with IANA - see RFC 2327
OtherTypeOfNetwork     = 1*(SuitableLCOCharacter)
supportedTypeOfNetwork = typeOfNetwork *(";" typeOfNetwork)

supportedModes    = ConnectionMode 0*(";" ConnectionMode)

supportedPackages = packageName 0*(";" packageName)

packageName = 1*(ALPHA / DIGIT / HYPHEN) ; Hyphen neither first or last

LocalOptionExtensionName = VendorLCOExtensionName
                         / PackageLCOExtensionName
                         / OtherLCOExtensionName
VendorLCOExtensionName   = "x" ("+"/"-") 1*32(SuitableExtLCOCharacter)
PackageLCOExtensionName  = packageName "/"
                           1*32(SuitablePkgExtLCOCharacter)
; must not start with "x-" or "x+"
OtherLCOExtensionName    = 1*32(SuitableExtLCOCharacter)

LocalOptionExtensionValue = (1*(SuitableExtLCOValChar)
                                                / quotedString)
                            *(";" (1*(SuitableExtLCOValChar)
                                                / quotedString))

;Note: No "data" mode.
ConnectionMode = "sendonly" / "recvonly" / "sendrecv"
               / "confrnce" / "inactive" / "loopback"
```

```
               / "conttest" / "netwloop" / "netwtest"
               / ExtensionConnectionMode
ExtensionConnectionMode = PkgExtConnectionMode
PkgExtConnectionMode    = packageName "/" 1*(ALPHA / DIGIT)

RequestedEvents = requestedEvent 0*("," 0*(WSP) requestedEvent)
requestedEvent  = (eventName ["(" requestedActions ")"])
                / (eventName "(" requestedActions ")"
                                      "(" eventParameters ")" )
eventName = [(packageName / "*") "/"]
                (eventId / "all" / eventRange
                                       / "*" / "#") ; for DTMF
                              ["@" (ConnectionId / "$" / "*")]
eventId = 1*(ALPHA / DIGIT / HYPHEN)   ; Hyphen neither first nor last
eventRange = "[" 1*(DigitMapLetter / (DIGIT "-" DIGIT) /
                        (DTMFLetter "-" DTMFLetter)) "]"
DTMFLetter = "A" / "B" / "C" / "D"

requestedActions = requestedAction 0*("," 0*(WSP) requestedAction)
requestedAction  = "N" / "A" / "D" / "S" / "I" / "K"
                 / "E" "(" EmbeddedRequest ")"
                 / ExtensionAction
ExtensionAction  = PackageExtAction
PackageExtAction = packageName "/" Action ["(" ActionParameters ")"]
Action           = 1*ALPHA
ActionParameters = eventParameters        ; May contain actions

;NOTE: Should tolerate different order when receiving, e.g., for NCS.
EmbeddedRequest = (      "R" "(" EmbeddedRequestList ")"
                    ["," 0*(WSP) "S" "(" EmbeddedSignalRequest ")"]
                    ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"]      )
                / (      "S" "(" EmbeddedSignalRequest ")"
                    ["," 0*(WSP) "D" "(" EmbeddedDigitMap ")"] )
                / (      "D" "(" EmbeddedDigitMap ")" )

EmbeddedRequestList   = RequestedEvents
EmbeddedSignalRequest = SignalRequests
EmbeddedDigitMap = DigitMap

SignalRequests   = SignalRequest 0*("," 0*(WSP) SignalRequest )
SignalRequest    = eventName [ "(" eventParameters ")" ]

eventParameters  = eventParameter 0*("," 0*(WSP) eventParameter)
eventParameter   = eventParameterValue
```

```
                / eventParameterName "=" eventParameter
                / eventParameterName "(" eventParameters ")"
eventParameterString = 1*(SuitableEventParamCharacter)
eventParameterName   = eventParameterString

eventParameterValue  = eventParameterString / quotedString

DigitMap           = DigitString  / "(" DigitStringList ")"
DigitStringList    = DigitString 0*( "|" DigitString )
DigitString        = 1*(DigitStringElement)
DigitStringElement = DigitPosition ["."]
DigitPosition      = DigitMapLetter / DigitMapRange
; NOTE "X" is now included
DigitMapLetter     = DIGIT / "#" / "*" / "A" / "B" / "C" / "D" / "T"
                   / "X" / ExtensionDigitMapLetter
ExtensionDigitMapLetter = "E" / "F" / "G" / "H" / "I" / "J" / "K"
                        / "L" / "M" / "N" / "O" / "P" / "Q" / "R"
                        / "S" / "U" / "V" / "W" / "Y" / "Z"
; NOTE "[x]" is now allowed
DigitMapRange = "[" 1*DigitLetter "]"
DigitLetter   = *((DIGIT "-" DIGIT) / DigitMapLetter)

ObservedEvents = SignalRequests

EventStates    = SignalRequests

ConnectionParameters = ConnectionParameter
                       0*( "," 0*(WSP) ConnectionParameter )

ConnectionParameter  = ( "PS" "=" packetsSent )
                     / ( "OS" "=" octetsSent )
                     / ( "PR" "=" packetsReceived )
                     / ( "OR" "=" octetsReceived )
                     / ( "PL" "=" packetsLost )
                     / ( "JI" "=" jitter )
                     / ( "LA" "=" averageLatency )
                     / ( ConnectionParameterExtensionName
                             "=" ConnectionParameterExtensionValue )
packetsSent     = 1*9(DIGIT)
octetsSent      = 1*9(DIGIT)
packetsReceived = 1*9(DIGIT)
octetsReceived  = 1*9(DIGIT)
packetsLost     = 1*9(DIGIT)
jitter          = 1*9(DIGIT)
```

```
averageLatency  = 1*9(DIGIT)

ConnectionParameterExtensionName = VendorCPExtensionName
                                 /    PackageCPExtensionName
VendorCPExtensionName  = "X" "-" 2*ALPHA
PackageCPExtensionName = packageName "/" CPName
CPName = 1*(ALPHA / DIGIT / HYPHEN)
ConnectionParameterExtensionValue = 1*9(DIGIT)

MaxMGCPDatagram = 1*9(DIGIT)

ReasonCode = 3DIGIT
             [1*(WSP) "/" packageName]     ; Only for 8xx
             [WSP 1*(%x20-7E)]

SpecificEndpointID = endpointName
SecondEndpointID   = endpointName

RequestedInfo = infoCode 0*("," 0*(WSP) infoCode)

infoCode = "B" / "C" / "I" / "N" / "X" / "L" / "M" / "R" / "S"
         / "D" / "O" / "P" / "E" / "Z" / "Q" / "T" / "RC" / "LC"
         / "A" / "ES" / "RM" / "RD" / "PL" / "MD" / extensionParameter

QuarantineHandling = loopControl / processControl
                   / (loopControl "," 0*(WSP) processControl )
loopControl    = "step" / "loop"
processControl = "process" / "discard"

DetectEvents = SignalRequests

RestartMethod = "graceful" / "forced" / "restart" / "disconnected"
              / "cancel-graceful" / extensionRestartMethod
extensionRestartMethod = PackageExtensionRM
PackageExtensionRM     = packageName "/" 1*32(ALPHA / DIGIT / HYPHEN)
RestartDelay = 1*6(DIGIT)

extensionParameter = VendorExtensionParameter
                   / PackageExtensionParameter
                   / OtherExtensionParameter
VendorExtensionParameter  = "X" ("-"/"+") 1*6(ALPHA / DIGIT)
PackageExtensionParameter = packageName "/"
                            1*32(ALPHA / DIGIT / HYPHEN)
; must not start with "x-" or x+"
```

```
OtherExtensionParameter   = 1*32(ALPHA / DIGIT / HYPHEN)

;If first character is a double-quote, then it is a quoted-string
parameterString = (%x21 / %x23-7F) *(%x20-7F) ; first and last must not
                                              ; be white space
                    / quotedString

MGCPResponse = MGCPResponseLine 0*(MGCPParameter)
                                        *2(EOL *SDPinformation)

MGCPResponseLine = responseCode 1*(WSP) transaction-id
                        [1*(WSP) "/" packageName]    ; Only for 8xx
                             [WSP responseString] EOL

responseCode = 3DIGIT
responseString = *(%x20-7E)

SuitablePkgExtLCOCharacter =  SuitableLCOCharacter

SuitableExtLCOCharacter = DIGIT / ALPHA / "+" / "-" / "_" / "&"
              / "!" / "'" / "|" / "=" / "#" / "?"
              / "." / "$" / "*" /       "@" / "[" / "]"
              / "^" / "`" / "{" / "}" / "~"

SuitableLCOCharacter   = SuitableExtLCOCharacter / "/"

SuitableExtLCOValChar  = SuitableLCOCharacter / ":"

; VCHAR except """, "(", ")", ",", and "="
SuitableEventParamCharacter = %x21 / %x23-27 / %x2A-2B
                            / %x2D-3C / %x3E-7E

; NOTE: UTF8 encoded
quotedString  = DQUOTE 0*(quoteEscape / quoteChar) DQUOTE
quoteEscape   = DQUOTE DQUOTE
quoteChar = (%x00-21 / %x23-FF)

EOL = CRLF / LF

HYPHEN = "-"

; See RFC 2327 for proper SDP grammar instead.
SDPinformation = SDPLine CRLF *(SDPLine CRLF)        ; see RFC 2327
SDPLine        = 1*(%x01-09 / %x0B / %x0C / %x0E-FF) ; for proper def.
```